

# Table Of Contents

---

<b>Table Of Contents</b> .....	i
--------------------------------	---

<b>Introduction</b> .....	1
---------------------------	---

<b>MEMS Pro System</b> .....	2
------------------------------	---

Total Solution .....	3
----------------------	---

<b>Tool Flow</b> .....	5
------------------------	---

Schematic Capture (S-Edit) .....	6
----------------------------------	---

Simulator (T-Spice Pro) .....	6
Layout Editor (L-Edit) .....	7
User-programmable Interface .....	8
Layout vs. Schematic (LVS) .....	9
3D Modeler .....	9
MEMS Block Place and Route .....	9
MEMS Library (MEMSLib) .....	10
Foundry Support .....	10
Embedded features in ANSYS .....	10
<i>Reduced Order Modeling (Macro-Model Generation)</i> .....	10
<i>Automatic mask layout generation (3D to Layout)</i> .....	11
<b>What's New in Version 3.0</b> .....	12
<b>Documentation Conventions</b> .....	13

<b>MEMS Pro Tutorial</b>	16
<b>Introduction</b>	17
The Design Example	18
<b>Creating a Schematic</b>	19
Launching S-Edit	19
Opening the File	20
Creating a New Module	22
Instantiating Components	23
<i>Instantiating a Plate</i>	23
<i>Instantiating Comb-drives</i>	24
<i>Instantiating Folded Springs</i>	25
Wiring Objects	26
<i>Zooming the View</i>	26
Instantiating Voltage Sources	29

Placing Global Nodes .....	29
Editing Object Properties .....	32
Labeling Nodes .....	34
Adding Simulation Commands .....	35
<b>Exporting a Netlist</b> .....	40
Tutorial Breakpoint .....	40
<b>Simulating from a Netlist</b> .....	42
Simulating with T-Spice .....	42
<b>Probing a Waveform</b> .....	45
<b>Viewing a Waveform</b> .....	47
Chart Setups .....	47
Trace Manipulation .....	48
<b>Generating a Layout</b> .....	56
Tutorial Breakpoint .....	56
Launching L-Edit .....	56

Opening the File .....	57
Creating Components .....	57
<i>Using the MEMS Library Palette</i> .....	58
<i>Generating the Plate</i> .....	59
<i>Generating the Comb-drives</i> .....	60
<i>Editing an Already Generated Component</i> .....	61
<i>Attaching Components</i> .....	63
<i>Generating the Folded Springs</i> .....	65
<i>Generating the Ground Plate</i> .....	66
<i>Generating the Bonding Pads</i> .....	67
Viewing Properties .....	70
<b>Viewing a 3D Model</b> .....	73
Tutorial Breakpoint .....	73
Launching L-Edit and Opening a File .....	73
Process Definition .....	74

<i>Importing the Process Definition</i> .....	74
3D Model View .....	78
<i>Generating the 3D Model</i> .....	78
<i>Manipulating the 3D Model View</i> .....	78
<i>Multiple Views</i> .....	80
<i>Viewing the 3D Model</i> .....	82
3D Cross-section .....	82
<b>Drawing Tools</b> .....	86
Tutorial Breakpoint .....	86
Drawing a Wire .....	88
Drawing a Torus .....	89
Drawing a Curved Polygon .....	91
Drawing a Circle .....	95
Drawing a Box .....	95

<b>MEMS Pro Toolbar</b> .....	97
<b>Introduction</b> .....	98
<b>Library Menu</b> .....	103
Library Palette .....	103
Edit Component .....	105
<b>3D Tools Menu</b> .....	109
Editing a Process Definition .....	109
Viewing a 3D Model .....	111
Deleting a 3D Model .....	112
Exporting a 3D Model .....	113
<b>Easy MEMS Menu</b> .....	114
Creating holes in a plate .....	114
Copying objects .....	116
<b>Splines</b> .....	118

Creating Splines .....118

Editing Splines .....120

**Tools** .....121

Viewing Vertex Coordinates .....121

Viewing Vertex Angles .....122

Viewing Vertex Information .....122

Clearing Vertex Information .....123

**Help** .....124

MEMS Pro User Guide .....124

About MEMS Pro .....124

**Splines** .....126

**Introduction** .....127



Understanding Splines .....	127
<b>Create Spline Dialog Box .....</b>	<b>129</b>
<b>Creating Splines .....</b>	<b>132</b>
Creating Splines from Angled Wires .....	132
<i>Interpolation</i> .....	134
<i>Approximation</i> .....	137
Re-creating Angled Wires .....	139
Creating Splines from Polygons .....	141
<b>Editing Splines .....</b>	<b>146</b>

<b>MEMS Pro Utilities .....</b>	<b>147</b>
---------------------------------	------------

<b>Introduction .....</b>	<b>148</b>
<b>Running Macros in L-Edit .....</b>	<b>149</b>

Loading the Macros .....	149
<b>Generating Polar Arrays .....</b>	<b>150</b>
Description .....	150
Accessing the Function .....	150
Parameters .....	152
<b>Generating Holes in a Plate .....</b>	<b>154</b>
<b>Viewing Vertex Coordinates and Angles ....</b>	<b>157</b>
Viewing Vertex Coordinates .....	157
Viewing Vertex Angles .....	159
Viewing Vertex Information .....	161
Clearing Vertex Information .....	163
<b>Approximating All-angle Objects .....</b>	<b>164</b>
Description .....	164
Accessing the Macro .....	164
Parameters .....	167

<b>Generating Concentric Circles</b>	168
Location	168
Description	168
Accessing the Macro	169
Parameters	169
Input File Format	169
<i>Syntax</i>	170
<i>Example</i>	170
Parameters	171

<b>3D Modeler</b>	172
-------------------	-----

<b>Introduction</b>	173
MCNC MUMPs Thermal Actuator	173

MCNC MUMPs Rotary Motor .....	175
Analog Devices iMEMS ADXL Accelerometer .....	177
Bulk Micromachined Diaphragm .....	180
<b>Accessing 3D Models .....</b>	<b>182</b>
3D Model Input .....	182
3D Modeler Output .....	182
Accessing the 3D Tools .....	184
<b>Defining Colors for 3D Models .....</b>	<b>186</b>
<b>Viewing 3D Models from Layout .....</b>	<b>188</b>
<b>3D Model View User Interface .....</b>	<b>190</b>
Application Elements .....	190
Title Bar .....	191
Menu Bar .....	192
<i>File Menu</i> .....	192
<i>View Menu</i> .....	195

<i>Tools Menu</i> .....	204
<i>Setup Menu</i> .....	205
<i>Window Menu</i> .....	206
<i>Help Menu</i> .....	207
3D Model Tool Bar .....	209
Palette .....	210
Status Bar .....	213
<b>Viewing a Cross-section</b> .....	215
<b>Deleting 3D Models</b> .....	218
<b>Exporting 3D Models</b> .....	220
<b>Linking to ANSYS</b> .....	223
<b>Editing the Process Definition</b> .....	225
Importing the Process Definition .....	226
Process Identification .....	228
Editing the Process Steps List .....	228

<i>Enable</i> .....	229
<i>Display 3D model for this step</i> .....	229
<i>Move Step</i> .....	229
<i>Add Step</i> .....	230
<i>Delete Step</i> .....	230
Editing Individual Process Steps .....	230
Wafer .....	231
Deposit .....	234
<i>DepositType = CONFORMAL</i> .....	235
<i>DepositType = SNOWFALL</i> .....	237
<i>DepositType = FILL</i> .....	239
Etch .....	241
<i>Orientation Considerations</i> .....	242
<i>EtchType = SURFACE</i> .....	244
<i>EtchType = BULK</i> .....	246

<i>EtchType = SACRIFICIAL</i> .....	248
MechanicalPolish .....	249
<b>3D Modeler Error Checks</b> .....	252
<i>Checking if the 3D Model is Out-of-Date</i> .....	253
<i>Checking if a Process Definition is used</i> .....	253
<i>Checking for Process with Derived Layers</i> .....	254
<i>Checking for the Existence of all Required Layers</i> ..	254
<i>Checking for Wires or Self-Intersecting Polygons</i> ..	254

<b>ANSYS Tutorial</b> .....	256
-----------------------------	-----

<b>Introduction</b> .....	257
Launching L-Edit .....	257
Opening the File .....	257

Viewing the 3D Model .....	258
Exporting the 3D Model .....	259
<b>Reading the 3D Model in ANSYS .....</b>	<b>261</b>
Viewing the 3D Model in ANSYS .....	261
Setting Material Properties .....	263
Adding an Element Type .....	264
<b>Setting Boundary Conditions .....</b>	<b>265</b>
<b>Meshing the Model .....</b>	<b>269</b>
<b>Running the Analysis .....</b>	<b>272</b>
<b>Displaying the Results .....</b>	<b>273</b>
<b>Computing the Spring Constant .....</b>	<b>276</b>
<b>Entering Models under Windows NT .....</b>	<b>277</b>

<b>ANSYS to Layout Generator .....</b>	<b>278</b>
--	------------



<b>Introduction</b>	279
<b>3-D to Layout Tools</b>	281
Overview	281
Import Memos	285
Creation of Volumes	287
Deletion of Volumes	291
Addition of Volumes	292
Component Names	292
Saving Memos	295
Unit	296
Exporting a CIF File	297
The LAYOUT Menu Item	298
<b>The Layout Generator Program</b>	299
<b>Definition of a Technology File</b>	303
<b>Limitations</b>	315

Negative Mask Without Hole .....	315
Substrate .....	315
Splines .....	315
Boolean Operations on Layers .....	315
<b>Tutorial</b> .....	316
Import Memos .....	316
3D Modifications .....	323
The Layout Generator Program .....	326

## Reduced Order Modeling .....334

<b>User Manual</b> .....	335
Introduction .....	335
R.O.M. Menu .....	337

Condensation Algorithm .....	340
<i>Fundamentals</i> .....	340
<i>Running the Condensation</i> .....	341
Reduction of Electrostatically Coupled Structural Systems	348
<i>Fundamentals</i> .....	349
<i>Running the Reduction Algorithm</i> .....	353
<b>ROM Tutorial</b> .....	370
Condensation: Reduction with Single DOF & Load Cases	370
<i>Model Generation</i> .....	372
<i>Performing Reduction</i> .....	374
Condensation: Reduction with Multiple DOFs & Load Cases	380
<i>Model Generation</i> .....	382
<i>Performing Reduction</i> .....	385
<i>Simulating a reduced model using the SPICE simulator</i>	392

## Reduction of Electrostatically Coupled Structural Systems 408

*Model Generation* .....410

*Performing Reduction* .....411

*Simulating a reduced model using the SPICE simulator* 422

## Optimization Tutorial .....436

**Introduction** .....437

**Setting up the Optimization** .....439

**Running the Optimization** .....453

**Examining the Output** .....454

# Verification .....457

**Introduction .....458**

**Adding Connection Ports .....459**

**Extracting Layout .....463**

**Extracting Schematic for LVS .....468**

**Comparing Netlists .....471**

# Command Tool .....474

**Introduction .....475**

Usage in S-Edit .....475

*Schematic Mode .....475*

*Symbol Mode .....476*

<i>Property Creation</i> .....	476
<b>Accessing the Command Tool</b> .....	477
Schematic Tools Toolbar .....	477
Module Menu .....	478
<b>Command Tool Dialog</b> .....	479
<b>Schematic Object Creation</b> .....	482
<i>Template Module</i> .....	482
<b>Symbol Mode</b> .....	483
Schematic Object in Symbol Mode .....	483
Create Property Dialog .....	484

## **Block Place and Route Tutorial** ....486

<b>Initializing the Design</b> .....	488
--------------------------------------	-----

<b>Routing the Design .....</b>	<b>499</b>
---------------------------------	------------

<b>Extending the MEMS Library .....</b>	<b>506</b>
---	------------

<b>Introduction .....</b>	<b>507</b>
---------------------------	------------

<b>Schematic Symbols .....</b>	<b>508</b>
--------------------------------	------------

<b>SPICE Models .....</b>	<b>511</b>
---------------------------	------------

Application Example .....	512
---------------------------	-----

<b>Layout Generators .....</b>	<b>514</b>
--------------------------------	------------

Sample Layout Generator .....	514
-------------------------------	-----

<b>MEMSLib Reference .....</b>	<b>518</b>
--------------------------------	------------

<b>Introduction</b>	519
Acknowledgment	522
<b>Using the MEMS Library</b>	524
<b>Accessing the MEMS Library Palette</b>	526
Show Details Button	528
Editing the Generated Layout Parameters	530
<b>Active Elements</b>	532
Linear Electrostatic Comb Drive Elements (S_LCOMB_1, S_LCOMB_2)	532
Linear Electrostatic Comb Drive Elements	534
Linear Side Drive Elements (S_LSDM_1, S_LSDM_2)	535
Linear Side Drive Elements	537
Unidirectional Rotary Comb Drive Elements - Type 1 (S_RCOMBU_1, S_RCOMBU_2)	538
Unidirectional Rotary Comb Drive Elements-Type1	541



Unidirectional Rotary Comb Drive Elements - Type 2	
(S_RCOMBUA_1, S_RCOMBUA_2)	542
Unidirectional Rotary Comb Drive Elements - Type2	.545
Bidirectional Rotary Comb Drive Elements (S_RCOMBD_1,	
S_RCOMBD_2)	546
Bidirectional Rotary Comb Drive Elements .....	549
Rotary Comb Drive Elements (S_RCDM_1, S_RCDM_2)	550
Rotary Comb Drive Elements .....	553
Rotary Side Drive Elements (S_RSDM_1, S_RSDM_2)	554
Rotary Side Drive Elements .....	556
Harmonic Side Drive Elements (S_HSDM_1, S_HSDM_2)	557
Harmonic Side Drive Elements .....	559
<b>Passive Elements</b> .....	560
Journal Bearing Elements 1 (S_JBEARG_1)	560
Journal Bearing Elements 1 .....	562

Journal Bearing Elements 2 (S_JBEARG_2) .....	563
Journal Bearing Elements 2 .....	565
Linear Crab Leg Suspension Elements - Type 1 (S_LCLS_1, S_LCLS_2) .....	566
Linear Crab Leg Suspension Elements - Type 1 .....	568
Linear Crab Leg Suspension Elements - Type 2 (S_LCLSB_1, S_LCLSB_2) .....	569
Linear Crab Leg Suspension Elements - Type 2 .....	571
Linear Folded Beam Suspension Elements (S_LFBS_1, S_LFBS_2) .....	572
Linear Folded Beam Suspension Elements .....	574
Dual Archimedean Spiral Spring Elements (S_SPIRAL_1, S_SPIRAL_2) .....	575
Dual Archimedean Spiral Spring Elements .....	577
<b>Test Elements</b> .....	578

Area-Perimeter Dielectric Isolation Test Structure Element (S_APTEST_1)	578
Area-Perimeter Dielectric Isolation Test Structure Element	580
Crossover Test Structure Element - Type 1 (S_COTEST_1)	581
Crossover Test Structure Element - Type 1 .....	583
Crossover Test Structure Element - Type 2 (S_COTEST_2)	584
Crossover Test Structure Element - Type 2 .....	586
Euler Column (Doubly Supported Beam) Elements (S_EUBEAM_1, S_EUBEAM_2)	587
Euler Column (Doubly Supported Beam) Elements .....	589
Array of Euler Column Elements (S_EUBEAMS_1, S_EUBEAMS_2)	590
Array of Euler Column Elements .....	592
Guckel Ring Test Structure Elements (S_GURING_1, S_GURING_2)	593

Guckel Ring Test Structure Elements .....	595
Array of Guckel Ring Elements (S_GURINGS_1) .....	596
Array of Guckel Ring Elements .....	598
Multilayer Pad Element (S_PAD_1) .....	599
Multilayer Pad Element .....	600
<b>Resonator Elements</b> .....	601
Plate (S_PLATE_1) .....	601
Plate .....	603
Comb Drive (S_LCOMB_3) .....	604
Comb Drive (comb) .....	606
Folded Spring (S_LFBS_3) .....	607
Folded Spring .....	609
Ground Plate (S_GDPLATE_1 .....	610
Ground Plate .....	611
Bonding Pad (S_PAD_2) .....	612

Bonding Pad .....	613
-------------------	-----

<b>Technology Setup .....</b>	<b>614</b>
-------------------------------	------------

<b>Introduction .....</b>	<b>615</b>
---------------------------	------------

<b>MCNC MUMPs .....</b>	<b>616</b>
-------------------------	------------

Device Examples .....	618
-----------------------	-----

<b>Analog Devices/MCNC iMEMS .....</b>	<b>619</b>
--	------------

<b>Sandia ITT .....</b>	<b>620</b>
-------------------------	------------

<b>MOSIS/CMU .....</b>	<b>621</b>
------------------------	------------

<b>MOSIS/NIST .....</b>	<b>622</b>
-------------------------	------------

<b>Process Definition</b>	623
<b>Introduction</b>	624
<b>Process Steps</b>	628
ProcessInfo	628
<i>Syntax</i>	628
<i>Example</i>	629
<i>Description</i>	629
Wafer	630
<i>Syntax</i>	630
<i>Example</i>	630
<i>Description</i>	631
Deposit	634
<i>Syntax</i>	634
<i>Example</i>	634

<i>Description</i> .....	635
<i>DepositType</i> = <i>CONFORMAL</i> .....	636
<i>Thickness and Scf</i> .....	639
<i>DepositType</i> = <i>SNOWFALL</i> .....	642
<i>DepositType</i> = <i>FILL</i> .....	644
Etch .....	647
<i>Syntax</i> .....	647
<i>Example</i> .....	647
<i>Description</i> .....	648
<i>Orientation Considerations</i> .....	649
<i>EtchType</i> = <i>SURFACE</i> .....	651
<i>EtchType</i> = <i>BULK</i> .....	654
<i>EtchType</i> = <i>SACRIFICIAL</i> .....	657
MechanicalPolish .....	659
<i>Syntax</i> .....	659

<i>Example</i> .....	659
<i>Description</i> .....	660
ImplantDiffuse .....	664
<i>Syntax</i> .....	664
<i>Example</i> .....	664
<i>Description</i> .....	665
<i>Orientation Considerations</i> .....	666
Grow .....	669
<i>Syntax</i> .....	669
<i>Example</i> .....	669
<i>Description</i> .....	670
<b>Editing the Process Definition</b> .....	675
<b>Process Definition Example: MUMPs</b> .....	676



**INDEX** .....683



# 1 Introduction

---

▪ MEMS Pro System	2
▪ Tool Flow	5
▪ What's New in Version 3.0	12
▪ Documentation Conventions	13

# MEMS Pro System

The interdisciplinary nature of Micro-Electro-Mechanical Systems (MEMS) and the expertise required to develop the technology is a significant bottleneck in the timely design of new products incorporating MEMS technology.

This issue calls for a new generation of design tools that combines aspects of EDA and mechanical / thermal / fluidic / optical / magnetic CAD.

MEMSCAP approach to solving this design bottleneck is based on the following principles and features:

- Supporting multiple flows: for the component engineer, multi-physics circuit designer and for the system engineer;
- Allowing data exchange between the different description levels: the structural level (FEM/BEM), the system/behavioral level (SPICE, HDL-A, VHDL-AMS, Verilog-AMS), and the physical level (mask layout);
- Targeting key features for MEMS specific design.

MEMS Pro, in combination with ANSYS Multiphysics or other 3D analysis programs, enables system designers, MEMS circuit designers, IC designers, process engineers, MEMS specialists, and packaging engineers to share critical design and process information in the most relevant language for each contributor.

The MEMS Pro package includes a schematic entry tool, an analog and mixed analog / digital circuit level behavioral simulator, a statistical analyzer, an optimizer, a waveform viewer, a full-custom mask-level layout editing tool, an automatic layout generator, an automatic standard cell placement and routing tool, a design rule checking feature, an automatic netlist extraction tool (from layout or schematic), a comparison tool between netlists extracted from layout and schematics (Layout Versus Schematic), and libraries of MEMS examples.

MEMS specific key features also available in MEMS Pro encompass a 3D solid model generator using mask layout and process information, a 3D solid model viewer with cross-section capability, a process description editor, true curved drawing tools and automation of time consuming tasks using the MEMS Pro Easy MEMS features.

Embedded features within ANSYS 5.6 allow automatic mask layout generation from an ANSYS 3D Model and process description (3D to Layout), as well as automatic MEMS behavioral Model generation in hardware description languages (Reduced Order Modeling).

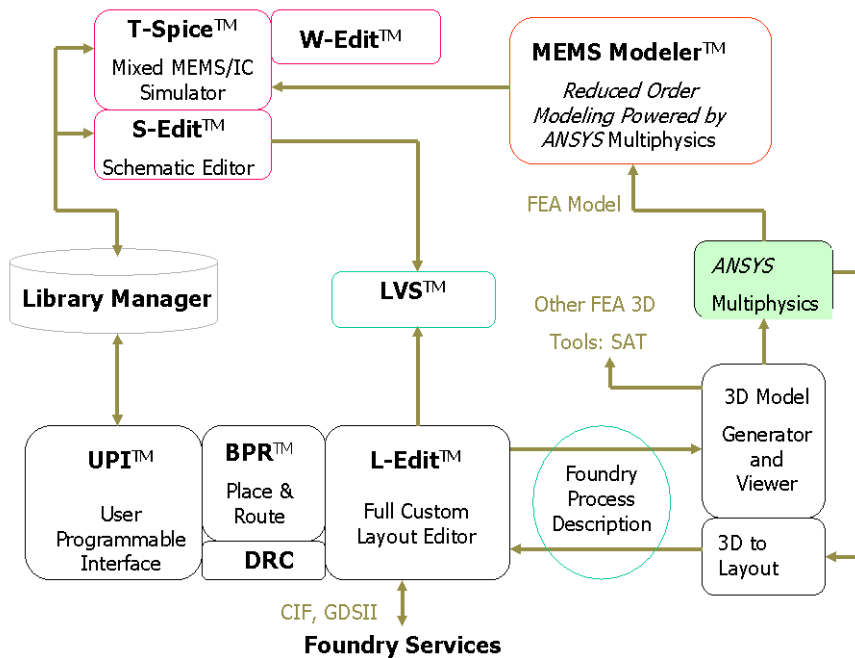
## Total Solution

S-Edit schematics are easily transferred to EDIF, SPICE and VHDL industry formats. L-Edit layout exports to standard formats accepted by mask makers and foundries including GDS II, CIF and DXF (through an optional converter). 3D

models also can be generated from layout and process definition and viewed in the L-Edit environment. These 3D models may also be exported in SAT format and directly used with third party tools such as AutoCAD, ANSYS, Ansoft HFSS, Maxwell 3D, ABAQUS, and MSC/NASTRAN and those from CFDRC and Coyote Systems. Now, Version 3.00 offers the possibility to generate CIF format layouts from 3D models. The Reduced Order Modeling feature generates MEMS behavioral models in SPICE and HDL-A for fast and accurate system level simulations.

# Tool Flow

Each stage of the MEMS design process is addressed by a different component of the MEMS Pro tool suite.



## Schematic Capture (S-Edit)

S-Edit is a fully hierarchical schematic capture program for MEMS and IC applications. The program also serves as a schematic entry front end to the T-Spice simulator, L-Edit/ SPR automatic standard cell placement and router, and layout vs. schematic (LVS) netlist comparison programs. S-Edit and its associated libraries are technology independent; that is, the design may be built and tested before choosing a specific manufacturing technology and vendor. User-defined global symbols convey connection among nodes without wiring. S-Edit also supports global node naming so that a single symbol can represent several distinct nodes in the design. Using S-Edit, MEMS schematics can be designed to include signals in multiple energy domains. For example, the MEMS Library includes a set of examples of electro-mechanical schematic symbols and models.

## Simulator (T-Spice Pro)

The T-Spice simulator provides full-chip analysis of analog, mixed analog/digital and MEMS designs using an extremely fast simulation engine that has been proven in designs of over 300,000 devices. For large circuits, the T-Spice simulator can be ten times faster than typical SPICE simulators.

MEMS macromodels can be implemented in 3 different ways in T-Spice. In the simplest form, MEMS devices may be modeled using equivalent circuits of standard SPICE components. Another method is to create table models from experimental data or finite element or boundary element analysis of the MEMS

devices. A third method is to use the external functional model interface. This last method allows quick and easy prototyping of custom MEMS macromodels using a C code interface.

The program includes standard SPICE models, BSIM3 models, and the advanced Maher/Mead charge-controlled MOSFET model that is ideally suited to sub-micron design. The W-Edit graphical waveform viewer, embedded within T-Spice, displays analysis results, and automatically updates its display each time T-Spice simulates a circuit.

Powerful optimization algorithms automatically determine device or process parameters that will optimize the performance of your design. Defining parameters to be varied, setting up optimization criteria, and choosing optimization algorithms is a cinch using the new Optimization Wizard. The Wizard prompts you for the optimization criteria the program will need.

Monte Carlo analysis generates “random” variations in parameter values by drawing them probabilistically from a defined distribution. This type of statistical analysis may be used to discover what effects process variation will have on system performance.

## Layout Editor (L-Edit)

L-Edit is an interactive, graphical layout editor for MEMS and IC design. This full-custom editor is fast, easy-to-use, and fully hierarchical. Primitives include boxes, polygons, circles, lines, wires, labels, arcs, splines, and tori. Drawing



modes include 90°, 45°, and all-angle layout. Shortcuts are also available for quickly laying out circles, tori, pie slices, splines, and “curved polygons” with true curved edges. Designs created in L-Edit are foundry ready. The new MEMS Pro Toolbar in Version 3.00 gives access to MEMS specific design features. They gather the creation of splines, the display of vertex information, and the use of Easy MEMS features like the polar array feature and the plate release feature. It also includes access to the process definition graphical interface, 3D modeler and viewer, MEMS specific DRC, and MEMSLib (the MEMS library).

## User-programmable Interface

L-Edit/UPI is a powerful tool for automating, customizing, and extending L-Edit’s command and function set. The heart of L-Edit/UPI is the macro interface. Macros are user-programmed routines, written in the C language, that describe automated actions or sets of actions. Macros can be recognized by their **.dll** file extension. Complex, parameterized cell generation (for example, comb drives, rotary motors, gears, etc.) as well as simple but often used geometry (for example, bonding pads) can be implemented with a single key stroke. L-Edit/UPI includes a C language interpreter for reading macro code, eliminating the need for a system compiler. The program reads **.dll** files produced by the user, or from MEMSCAP or Tanner libraries, or libraries supplied by third party vendors. The UPI provides user access to L-Edit’s Design Rule Checker and netlist extraction modules, and may be used to integrate L-Edit with other third-party applications.

## Layout vs. Schematic (LVS)

LVS compares the SPICE netlist generated from S-Edit or another schematic editor with the netlist generated from layout by L-Edit/Extract. LVS is a check to ensure that both netlists represent the same multiphysics "circuit". Should any inconsistencies be found between the two lists, LVS can be used to identify and resolve the ambiguity.

## 3D Modeler

Accurate three-dimensional (3D) visualization of your design-in-progress is crucial to successful fabrication. You can create 3D models of your MEMS device layout geometry directly in L-Edit using one of the many foundry fabrication process descriptions we support, or by specifying your own custom process. The 3D Solid Modeler permits views of surface and bulk micromaching steps including deposit, etch, and mechanical polishing. You can easily customize your view with features such as panning, zooming, cross-section modeling, and other viewing controls. 3D solid model geometry can be exported in a SAT format.

## MEMS Block Place and Route

The block place and route feature will save you time and prevent wiring mistakes. Routing may be done automatically or manually. The MEMS block place and route enables you to connect component level blocks of MEMS and IC

devices. Efficiency enhancing features include hierarchical block placement, block level floor planning, an EDIF netlist reader, and on-line signal integrity analysis.

## **MEMS Library (MEMSLib)**

MEMSLib provides MEMS designers with schematics, simulation models, and parameterized layout generators for a set of MEMS components. MEMSLib includes several types of suspension elements, electro-mechanical transducers, and test structures for extracting material properties. Various example elements can be assembled to produce a single MEMS device.

## **Foundry Support**

We've included examples of process setup information for design rules, layer definitions, extraction rules, process definitions, model parameter values, and macros from the most popular foundries. Processes examples include MCNC (MUMPs), Sandia (M3M), ADI (iMEMS), and MOSIS (NIST).

## **Embedded features in ANSYS**

### *Reduced Order Modeling (Macro-Model Generation)*

Powered by ANSYS Multiphysics, MEMS Modeler offers automatic generation of behavioral models for fast and accurate system level simulation. It captures the

essential behavior for mechanical devices, and coupled electrostatics-mechanics MEMS components. Transient simulations, "what if" analysis and very accurate system simulation are then easily and quickly performed.

### *Automatic mask layout generation (3D to Layout)*

FEM-to-layout automatically generates mask layout in CIF format from FEM models developed from a target process definition.

## What's New in Version 3.0

MEMS Pro has been enhanced to simplify design flow and boost your productivity. We have incorporated technology that will let you optimize your designs before you submit them to the foundry, and thereby shorten your design cycle. For more information about

- The new MEMS-specific Graphical User Interface, refer to MEMS Pro Toolbar on page 97
- Easy MEMS including the plate release feature, the polar array feature, and the vertex information viewer, refer to MEMS Pro Utilities on page 147
- Automatic spline generator, refer to Splines on page 126
- 3D to Layout generator, refer to ANSYS to Layout Generator on page 278
- Reduced Order Modeling, refer to Reduced Order Modeling on page 334

# Documentation Conventions

This section contains information about the typographical and stylistic conventions employed by this user guide.

In-line references to menu and simulation commands, device statements, special characters, and examples of user input and program output are represented by a bold font. For example: **.print tran v(out)**.

Elements in hierarchical menu paths are separated by a > sign. For example, **File > Open** means the **Open** command in the **File** menu.

Tabs in dialog boxes are set off from the command name or dialog box title by a dash. For example, **Setup > Layers—General** and **Setup Layers—General** both refer to the **General** tab of the **Setup Layers** dialog.

Freestanding quotations of input examples, file listings, and output messages are represented by a constant-width font—for example:

```
.ac DEC 5 1MEG 100MEG
```

Variables for which context-specific substitutions should be made are represented by bold italics—for example, ***myfile.tdb***.

Sequential steps in a tutorial are set off with a check-box dingbat (☑) in the margin.

References to keyboard-mouse button combinations are given in boldface, with the first letter capitalized—for example, **Alt + Left**. The terms left-click, right-click, and center-click all assume default mappings for mouse buttons.

Text omitted for clarity or brevity is indicated by an ellipsis (...).

Special keys are represented by abbreviations, as follows.

<i><b>Key</b></i>	<i><b>Abbreviation</b></i>
Shift	<b>Shift</b>
Enter	<b>Enter</b>
Control	<b>Ctrl</b>
Alternate	<b>Alt</b>
Backspace	<b>Back</b>
Delete	<b>Del</b>
Escape	<b>Esc</b>
Insert	<b>Ins</b>
Tab	<b>Tab</b>

<i>Key</i>	<i>Abbreviation</i>
Home	<b>Home</b>
End	<b>End</b>
Page Up	<b>PgUp</b>
Page Down	<b>PgDn</b>
Function Keys	<b>F1 F2 F3 ...</b>
Arrow Keys	↓, ←, →, ↑

When certain keys are to be pressed *simultaneously*, their abbreviations are adjoined by a plus sign (+). For example, **Ctrl + R** means that the **Ctrl** and **R** keys are pressed at the same time.

When certain keys are to be pressed *in sequence*, their abbreviations are separated by a space ( ). For example, **Alt + E R** means that the **Alt** and **E** keys are pressed at the same time and then released, immediately after which the **R** key is pressed.

Abbreviations for *alternative* key-presses are separated by a slash (/). For example, **Shift + ↑ / ↓** means that the **Shift** key can be pressed together with either the up (↑) arrow key or the down (↓) arrow key.



## 2 MEMS Pro Tutorial

---

▪ Introduction	17
▪ Creating a Schematic	19
▪ Exporting a Netlist	40
▪ Simulating from a Netlist	42
▪ Probing a Waveform	45
▪ Viewing a Waveform	47
▪ Generating a Layout	56
▪ Viewing a 3D Model	73
▪ Drawing Tools	86

# Introduction

In the MEMS Pro tutorial, you will follow the complete design of a MEMS resonator. The ANSYS Tutorial on page 176, the ROM Tutorial on page 370, the ANSYS to layout Tutorial on page 316, the Block Place and Route Tutorial on page 486, and the Optimization Tutorial on page 435 are targeted to special features of MEMS Pro. The advanced portion of this tutorial includes a demonstration of layout extraction and netlist comparison using L-Edit/Extract and LVS. Those chapters assume that the user is completely familiar with the material covered in this general tutorial.

In this tutorial, you will create a schematic design, analyze system behavior, and generate device layout with the MEMS Pro tools S-Edit, T-Spice, W-Edit, and L-Edit. You will draw mask layout manually and automatically, and generate and view 3D models and cross-sections in L-Edit.

All files mentioned in this chapter are located in the **tutorial** subdirectory of the main MEMS Pro installation directory. We recommend that you follow the tutorial from the beginning; however, you may enter and exit the tutorial at several points during the design. Tutorial breakpoints occur at Simulating from a Netlist on page 42, Generating a Layout on page 56, Viewing a 3D Model on page 73, and at Drawing Tools on page 86.


## The Design Example

The design example, appearing throughout the *MEMS Pro User Guide*, is an electrostatic lateral comb-drive resonator. A resonator is a MEMS transducer that can be used as a sensor by exploiting the high sensitivity of its resonant frequency to various physical parameters. The resonator was chosen for your review because it is an easily understood coupled electro-mechanical system. The resonator will be designed using the MEMSLib library components including comb-drives, a plate and folded springs.

# Creating a Schematic

In this section, you will learn how to navigate and manipulate designs using the S-Edit schematic editor.

## Launching S-Edit

- ☑ Launch S-Edit by double-clicking the **S-Edit** icon  in the installation directory.

The S-Edit user interface consists of five areas:

- The *title bar* (at the very top of the application window) contains the file name, module name, page name, and mode.
- The *menu bar* (at the top) contains commands
- The *palette bar* (on the left) contains tool icons
- The *status bar* (at the bottom) contains runtime information
- The *display area* (in the center) contains the schematic
- The *standard commands toolbar* (below the menu bar) contains often used commands

Designs are contained in *files*, each of which consists of one or more *modules*. Modules are viewed in either of two modes:

- **Symbol Mode:** a graphical representation of the module showing the module's connections.
- **Schematic Mode:** shows the composition and connectivity of the module.

The schematic may contain one or more pages which consist largely of two component types:

- *Primitives:* geometrical objects, wires, ports, annotation objects, and labels; all created with the S-Edit drawing tools.
- *Instances:* “copies” of other modules, dynamically linked to their originals. Instances are displayed in a design using the module's symbol page.

### Note

---

For more information on *instancing modules*, see Working with Modules on page 97 of the *S-Edit User Guide and Reference*.

---

## Opening the File

As part of this section of the tutorial, you will place the components of an electrostatic lateral comb-drive resonator, connect those components, and run an AC analysis on your design.

The schematic symbols for these components have two pins for each connecting side: one carrying the electrical signal (denoted with the subscript **\_e**), and the other carrying the mechanical signal (denoted with the subscript **\_m**). These symbols are assembled to form the resonator design and a frequency sweep (AC analysis) of the system is performed to discover the resonant frequency and the magnitude of displacement. The electro-mechanical behavior of the components are modeled by expressing the mechanical behavior in terms of electrical analogs. These models can then be used to solve for the electrical and mechanical behavior of the system as well as the coupling between the two energy domains.

The completed design is provided for your reference in the **reson.sdb** file in the **tutorial** directory.



Select **File > Open** to open this file.

The current (visible) file, module, page, and mode are named at the top of the title bar. The schematic view of the resonator appears in Figure 1.

### Note

---

For more information, see Working with Files on page 92, Working with Modules on page 97, Working with Schematic Pages on page 116, Levels of Design on page 31 and Viewing Modes on page 33 of the *S-Edit User Guide and Reference*.

---

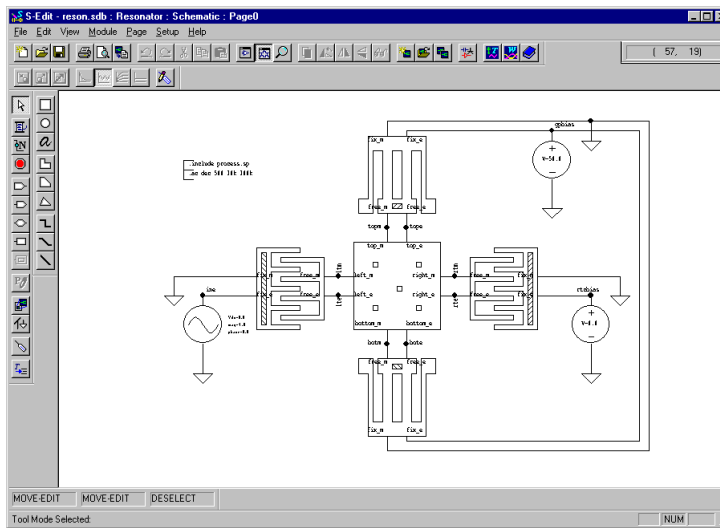


Figure 1: Schematic view of the complete resonator

## Creating a New Module

To initiate your new resonator design, you must first create a new module.

- ☑ Select **Module > New** to create a module.
- ☑ In the **Module Name** edit field, enter **MyResonator** and click **OK**.

Now would be a good time to save a copy of the file.

- ☑ Select **File > Save As** to invoke the **Save As** dialog.
- ☑ Select the **tutorial** directory, enter **myreson.sdb** as the filename, and click the **Save** button.

You can compare your work to the reference design at any time by using the **Module > Open** command and choosing **Resonator** as the module to open. Use **Module > Open** again to return to your design, this time selecting **MyResonator** as the module to be opened.

## Instantiating Components

### *Instantiating a Plate*

- ☑ Select **Module > Instance** to invoke the **Instance Module** dialog.
- ☑ Select **plate4** as the **Module to Instance** and click **OK**.

**Plate4**, a four-sided plate with eight points of connection (pins), will appear at the center of the schematic page.



- ☑ Home the view by selecting **View > Home** or by pressing the **Home** key. The view of the plate will be resized so that the plate fills the contents of the window.

### *Instantiating Comb-drives*

- ☑ Instantiate the **comb** module as you instantiated the plate.

The newly instantiated **comb** will appear on top of **plate4** in the middle of the schematic window. You will have to move it to a new location using the S-Edit click and drag feature.

#### **Note**

---

Objects in S-Edit can be moved by selecting with the left or right mouse button and dragging with the center mouse button. For two-button mice, press the **Alt** key and left-click to drag objects.

---

- ☑ Place the comb-drive to the right side of the previously instantiated plate.
- ☑ Place a second comb-drive into the design by copying the instance. With the first **comb** instance selected, select **Edit > Copy**, then **Edit > Paste**.
- ☑ Select the left **comb** and then flip it by choosing **Edit > Flip > Horizontal**.

- ☑ Move the comb-drives so that their connection pins, represented by circles, line up with the pins on the **plate4** instance (Figure 2) (see Pins on page 180 of the *S-Edit User Guide and Reference*).

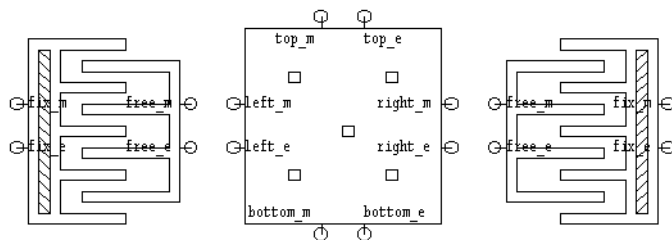


Figure 2: Aligning the comb-drives to the plate

### *Instantiating Folded Springs*

- ☑ Instantiate the **fspring** module and place it above the plate.
- ☑ Create a copy of the folded spring and place it below the plate.
- ☑ Flip this second instance of **fspring** by selecting **Edit > Flip > Vertical**.

## Wiring Objects

Wires are drawn using the **Wire** tool



First time, users of S-Edit may confuse the **Wire** tool with the **Line** tool. Lines are used to graphically represent components; they are non-electrical objects used to annotate your schematic. Wires are electrical and are used to connect objects.

### Note

---

For more information on *wiring your schematic*, see Wires on page 175 of the *S-Edit User Guide and Reference*.

---

## Zooming the View

Sensitive operations such as wiring nodes require a closer view for accuracy.

- ☒ Select **View > Zoom > Mouse**.
- ☒ Drag a box around the plate with the left mouse button. Allow enough room to see the areas between the comb-drives and the folded springs.

If you find you have zoomed in too much or too little, use the plus and minus keys to **Zoom** in and out. The arrow keys can be used to pan the view.

**Note**

---

For more information on *zooming*, see Panning and Zooming on page 134 of the *S-Edit User Guide and Reference*.

---

You will now create connections between the plate and other schematic components with wires.

- ☑ Select the **Wire** tool from the schematic toolbar.
- ☑ Initiate the wire placement by left-clicking on **plate4** at the **bottom\_m** pin. The pin is shown as an open circle on the bottom left of the **plate4** instance.

Vertices can be placed on wires by clicking the left mouse button while placing a wire.

**Note**

---

For more information on *making connections*, see Nodes on page 184, Pins on page 180, and Wires on page 175 of the *S-Edit User Guide and Reference*.

---

- ☑ Move the cursor down and end the wire placement by right-clicking at the pin called **free\_m** on the bottom **fspring**. This pin is shown as an open circle on the top left of the bottom **fspring** instance.
- ☑ Repeat this process to wire the plate with the other components (see Figure 3).

- ☒ Home the view by pressing the **Home** key.

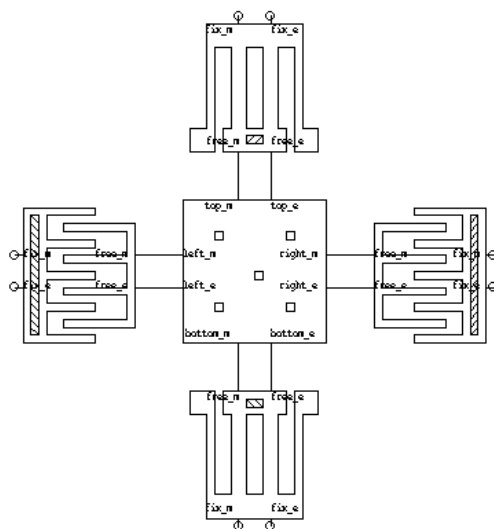


Figure 3: Schematic view of the wired elements

Next, you will add stimuli and commands to set up this schematic for simulation.

## Instantiating Voltage Sources

- ☑ Instantiate the **Source\_v\_ac** module.
- ☑ Place it to the left of the left **comb**.
- ☑ Instantiate the **Source\_v\_dc** module.
- ☑ Place it to the right of the right **comb**.
- ☑ Copy the instance of **Source\_v\_dc** and place it to the right of the top **fspring**.
- ☑ Wire the positive terminals of the voltage sources to the **fix\_e** pins of the right **comb**, left **comb**, and top **fspring**. The positive terminals are on the top of the voltage source, in this example.
- ☑ Compare your design to the finished design in Figure 1 to make sure you have placed the voltage sources correctly.

## Placing Global Nodes

Global nodes simplify the drawing and maintenance of schematics. They allow nodes throughout a design to be connected to each other without the need to draw or delete wires. Global nodes are especially useful for power, ground, anchor, clock, reset, and other system-wide nodes that require routing throughout the hierarchy of the design.

**Note**


---

For more information on *global nodes*, see Global Nodes on page 191 of the *S-Edit User Guide and Reference*.

---

To create a global node, you must place a global symbol on the design with the **Global Symbol** tool. Global symbols are special instances that function as wireless connectors. When you attach a node to a global symbol, you connect that node to all other nodes on every page and module in the design file that are attached to the same global symbol. Such nodes then become global nodes.

You will add six global ground symbols to the schematic. Three of them will be connected to the negative terminals of the voltage sources to set electrical grounds. The other three will be connected to the fixed mechanical terminals to signify mechanical anchors.

- ☒ To place a ground symbol onto the design, click the **Global Symbol** tool on the left side of the schematic window .

- ✓ Left-click on the schematic page. The **Instance Module** browse box will appear, with a list of the available global nodes and the ground (**Gnd**) symbol will be preselected.

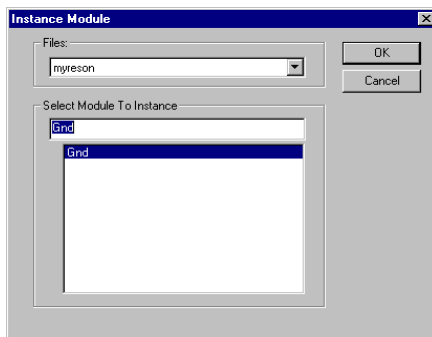


Figure 4: **Instance Module** browse box

- ✓ Click **OK**.
- The ground symbol will be placed where you left-clicked in the previous step.
- ✓ Copy and paste the ground symbol five times. Move two ground symbols to a place on the schematic near each voltage source.
  - ✓ Now wire the negative (lower) terminal of each of the three voltage sources to a ground symbol.



- ☑ Of the remaining three ground symbols, one should be connected to the **fix\_m** pin of the top spring, and the other two should be connected to the **fix\_m** pins of the two comb-drives.
- ☑ Compare your wiring to the completed schematic presented in Figure 1.
- ☑ Pins **fix\_e** (fixed electrical) and **fix\_m** (fixed mechanical) of the bottom **fspring** should be the only pins left unconnected at this point. Connect them to the **fix\_e** and **fix\_m** pins, respectively, of the top **fspring**.
- ☑ Compare your design to the finished design presented in Figure 1 to make sure the resonator has been wired correctly.

## Editing Object Properties

Now, you will edit the properties of one of the voltage sources in the schematic to set up the design for simulation.

- ☑ Select the voltage source next to the left **comb** by right-clicking it. Invoke the **Edit Instance of Module Source\_v\_ac** dialog by selecting **Edit > Edit Object**.

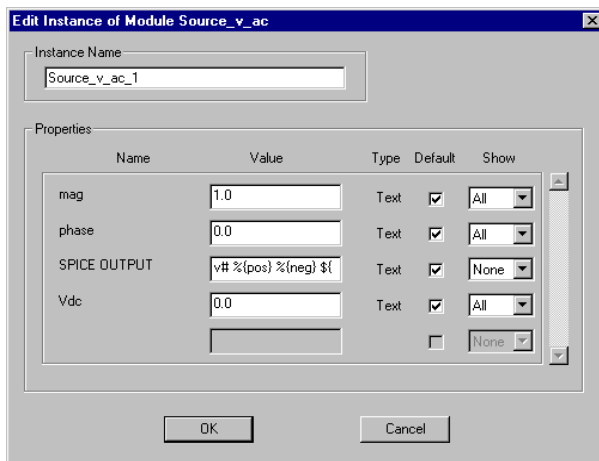



Figure 5: **Edit Instance of Module Source\_v\_ac** dialog

- ☒ Enter **1** for **mag**, **0** for **phase**, and **0** for **Vdc** in the corresponding edit fields.
- ☒ Click **OK**.
- ☒ Give the voltage source for the right comb-drive a DC value of 0 volts. Do this by choosing **Edit > Edit Object** with this source selected. Enter **0** in the **V** field.
- ☒ Similarly, give the voltage source for the folded springs a DC value of 50 Volts.

## Labeling Nodes

In S-Edit, connectivity is defined in terms of *nodes*. A node is a point on the schematic to which one or more pins or wires are connected. Nodes are defined by their name, and the *scope* of a node is normally the collection of schematic pages in a module. That is, if a node-name appears twice within a single module, both names refer to the same point of connection. If the same node-name appears within two different modules, the nodes refer to completely different points of connection. S-Edit automatically assigns names to each node, but you may also manually name nodes. User-assigned node labels are helpful for annotating S-Edit schematics and producing more readable netlists.

- ☒ Select the **Node Label** tool  from the schematic toolbar.
- ☒ Label the two wires connecting **plate4** to the right **comb**, **rtm**, and **rte**. To label a node, click it and enter the new node name in the **Place Node Label** dialog box. The **rtm** node label should be placed on the wire between **right\_m** and **free\_m** pins, and the **rte** node label should be placed on the wire between **right\_e** and **free\_e** pins.
- ☒ To change the orientation of the node label, click the **Selection** button, click the node label, and select **Edit > Edit Object**. From the **Edit Node Label** dialog box, click one of the eight radio buttons representing the location of the label origin.

- ☑ Edit the node label orientations to look somewhat like the layout in Figure 6.

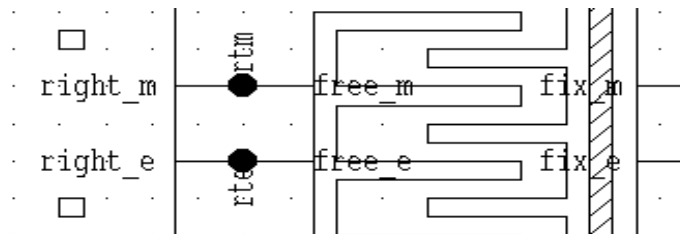



Figure 6: rtn and rte nodes

- ☑ You may rename the rest of the nodes in your diagram to match the names we have given ours in the **Resonator** module in **reson.sdb**, if you wish. This is an optional step. S-Edit will automatically assign names to unlabeled nodes.

## Adding Simulation Commands

The **Command** tool provides an easy, convenient means of entering device and model statements, stimuli, simulation commands, and simulation options within the S-Edit environment.

We will use the **Command** tool to add two SPICE commands. One instructs the simulator to run an AC simulation. The other instructs the simulator to include a file in the simulation netlist that contains fabrication process parameters for the resonator components.

- ☑ Select the **Command** tool  from the schematic toolbar.
- ☑ Click the work area to invoke the **T-Spice Command Tool** dialog (Figure 7).

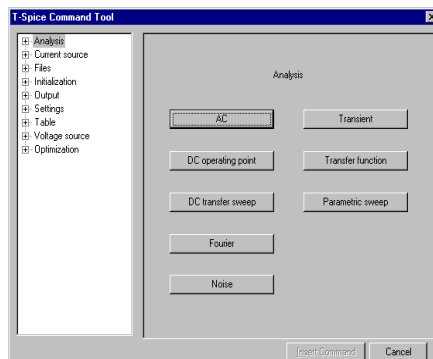


Figure 7: **T-Spice Command Tool** dialog

The **T-Spice Command Tool** dialog lists command categories on the left. By default, the **Analysis** category is selected and the right side of the dialog contains buttons listing the commands within that category. This command list may also be viewed by clicking the **+** sign next to each category. For example, clicking the **+** sign next to **Analysis** category will expand this category and show the same list of commands as those on the buttons. When a command is selected, the right side of the dialog changes to contain the parameters for the selected command.

- ☑ Add an AC analysis command by clicking the **AC** button on the right side of the **T-Spice Command Tool** dialog.

The directory tree on the left side of the **T-Spice Command Tool** dialog will open up to list the commands available under **Analysis**. The right side of the **T-Spice Command Tool** dialog will contain parameters specific to the **AC Analysis** command.

- ☑ Select **decade** as the **Frequency sampling type**, set **Frequencies per decade** to **500**, **Frequency range From** to **10k** and **Frequency range To** to **100k**. Click **Insert Command**.

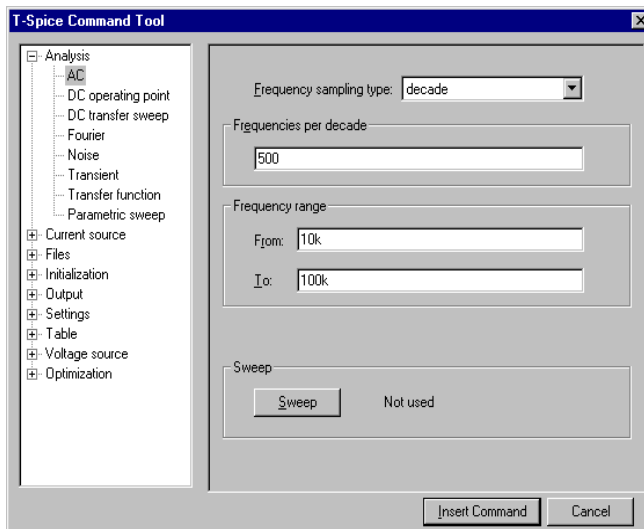


Figure 8: Customizing the AC analysis

Once the AC analysis is set up, we need to bring fabrication process information into the netlist. The steps below guide you through this task.

- ☑ Click the work area to open the **T-Spice Command Tool** dialog. Click the **+** next to the **Files** entry of the tree located on the left side of the dialog.
- ☑ Click **Include file** under **Files**.
- ☑ Set **Include file** to **process.sp** and click the **Insert Command** button. You can type the filename in, or you can find it with the **Browse** button.

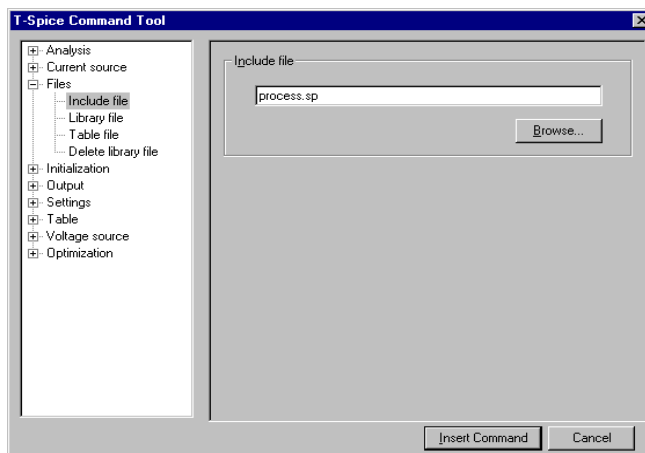


Figure 9: Selecting the technology process file



## Exporting a Netlist

An S-Edit schematic can be exported to a SPICE netlist by performing one of the following operations:

- Using the **Export Netlist** dialog box accessed via **File > Export**.
- Clicking the **T-Spice** button on the **Standard Commands** toolbar.


The netlist can be used to test the performance characteristics of the system using T-Spice or other SPICE programs.

The next few instructions ask you to invoke T-Spice from S-Edit to export a netlist and to run a simulation. When you invoke T-Spice, a new, active application window will appear. The current S-Edit window will become inactive, but *do not close it*. You will be returning to S-Edit to analyze your simulation results.

## Tutorial Breakpoint

You will now use T-Spice to simulate a circuit. If you are starting the tutorial here, double click the **S-Edit** icon and select **File > Open** to open the **reson.sdb** file in the **tutorial** directory.

Note that we have provided a working module of the resonator for you to use through the rest of the tutorial if the resonator you created is incomplete, or if you are entering the tutorial at this step. Our module is called **Resonator**. Follow the next two steps to access **Resonator**. If you want to use the resonator you have created, move ahead to the third step “Launch T-Spice.”

- ☑ Use the **Module > Open** command.
- ☑ Select the module **Resonator**, click **OK**. Click the page containing **Resonator** to ensure that it is active.
- ☑ Launch T-Spice. Click the **T-Spice** button  located in the **Standard Commands** toolbar. T-Spice will launch with the exported netlist open.

If you chose your resonator module, the exported netlist file name will be **MyResonator.sp**. If you chose our resonator module, the exported netlist file name will be **Resonator.sp** in the **tutorial** directory. This name will appear in the title bar of the input file window of T-Spice. You should leave S-Edit open.

### Note

---

For more information on *exporting schematics*, see Exporting a Netlist on page 228 of the *S-Edit User Guide and Reference*.

---

## Simulating from a Netlist

Using T-Spice and W-Edit, SPICE netlists can be simulated, and the simulation results can be displayed graphically. In this example, the coupled electro-mechanical behavior of the resonator is simulated using SPICE.

### Simulating with T-Spice

T-Spice contains a full featured editor that includes search and replace of strings and regular expressions, incremental find, and the **Command** tool for SPICE syntax assistance. There are four areas on the T-Spice user interface:

- The *menu bar* (at the top) contains menu commands
- The *toolbar* (beneath the menu bar) contains tool icons
- The *status bar* (at the bottom) contains runtime information
- The *work area* (in the center) contains input file and runtime information windows

The T-Spice window (Figure 10) allows to view a SPICE file in which the exported resonator components, their simulation parameters and connectivity are displayed.

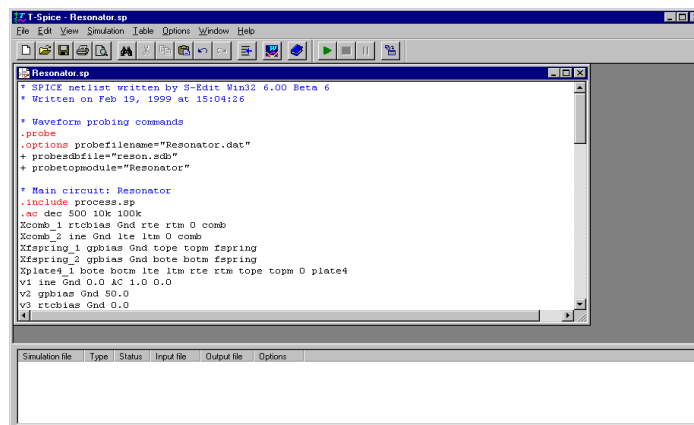



Figure 10: **T-Spice** window

- ☒ Click the **Run Simulation** button  located in the **Simulation** toolbar to run the simulation.
- ☒ Click **Start Simulation** in the **Run Simulation** dialog. The AC analysis that you set up from S-Edit will now be performed.

The **Simulation Output** window will appear, displaying simulation statistics and progress information, as well as any warning or error messages.


During the AC analysis, simulation results are recorded. Once the simulation is complete, you may examine the analysis results using the S-Edit **Probe** tool.

## Probing a Waveform

The waveform probe is used to browse through an S-Edit design and probe nodes to examine circuit simulation results for the specified node. When a node is probed, S-Edit invokes W-Edit, which automatically displays the waveform corresponding to the simulation results for that node. W-Edit can also be launched from T-Spice by selecting **Window > Show Waveform Viewer** or by clicking the **W-Edit** button from the **T-Spice** toolbar. In this tutorial, we will invoke W-Edit from S-Edit using the probing feature.

### Note

For more information on *waveform probing*, see Waveform Probing on page 243 of the *S-Edit User Guide and Reference*.


- ☑ Click somewhere in the S-Edit schematic window that contains the resonator schematic to re-activate S-Edit.
- ☑ Click the **Probe** tool  located on the **Schematic** toolbar. The cursor now has the shape of the **Probe** tool.
- ☑ Left-click with the **Probe** tool on the *rtm* node.

During waveform probing, W-Edit is launched, graphically displaying the results of the T-Spice simulation.

The W-Edit window should display a chart containing the magnitude and phase of the displacement of node ***rtm*** for the performed AC analysis.

## Viewing a Waveform

The W-Edit application window can contain many child windows, each containing one or more charts.

- ☑ Maximize the window containing your results by clicking the  button on the upper right corner of the window.

### Chart Setups

W-Edit allows the expansion of charts with more than one trace into separate charts, each containing a single trace. Collapsing the chart causes W-Edit to show all the visible traces in one chart.

- ☑ Select **Chart > Expand Chart**.

There should now be two charts (Figure 11), one with amplitude information for node **rtm**, **vm(rtm)**, and one showing the phase angle, **vp(rtm)** plotted versus frequency.





Now you should be able to view a peak in amplitude at around 13 kHz.

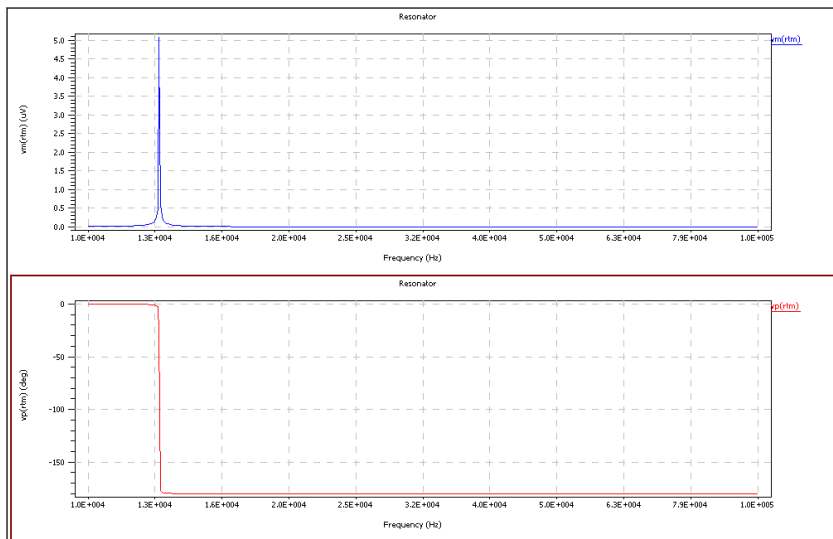


Figure 11: Charts representing the amplitude and the phase angle

## Trace Manipulation

At times, you may find it necessary to hide traces in order to simplify a waveform window.

- ☑ Select both charts by choosing **Edit > Select All**.
- ☑ Choose **Chart > Collapse Charts**.
- ☑ With the (now single) chart selected, choose **Chart > Traces**.
- ☑ In the **Traces** dialog, select **vp(rtm)**.
- ☑ Click the box beneath the **Show** label to *unselect* **vp(rtm)**. The checkmark will disappear.

The trace information will still be available, but the trace will not appear the next time you view the chart.

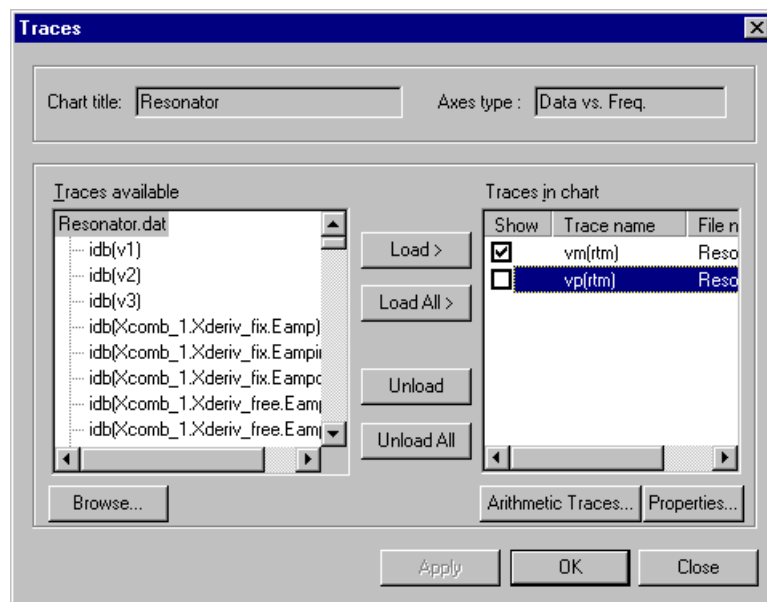


Figure 12: **Traces** dialog

- ☒ Click **Apply** and then **OK** to return to the chart.

Notice that the displayed unit for **vm(*rtm*)** is **Volts**. The mechanical behavior of this system is modeled with electrical analogs of mechanical components; the mechanical displacement maps to voltage. Therefore, **vm(*rtm*)** represents the displacement at the ***rtm*** node. Let's change the label on the dependent variable axis from **Volts** to **Displacement**. The new Y-axis units will be **meters**.

- ☑ Select the top chart. Choose **Chart > Options** to invoke the **Chart Options** dialog. Click the **Axes** tab. Enter **Displacement(rtm)** in the **Y-axis Label** field and **m** as the **Y-axis Units**.

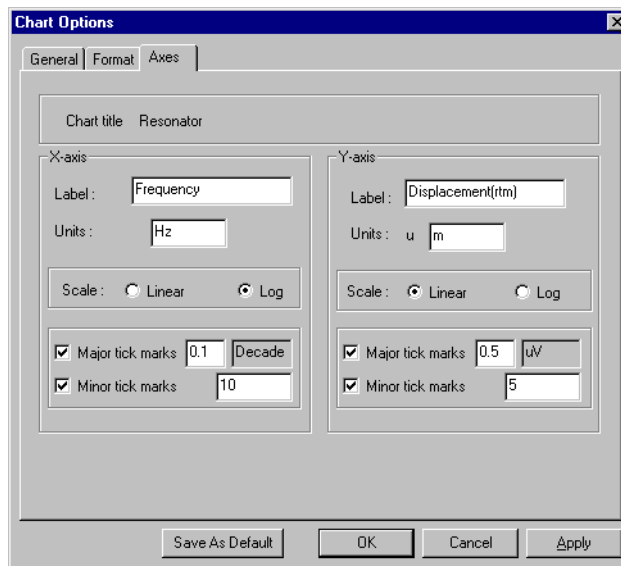


Figure 13: **Chart Options** dialog

The chart title can also be customized, if you wish.



In the **General** tab of the **Chart Options** dialog, set the **Chart title** to **Lateral Comb-drive Resonator** and click **OK**.

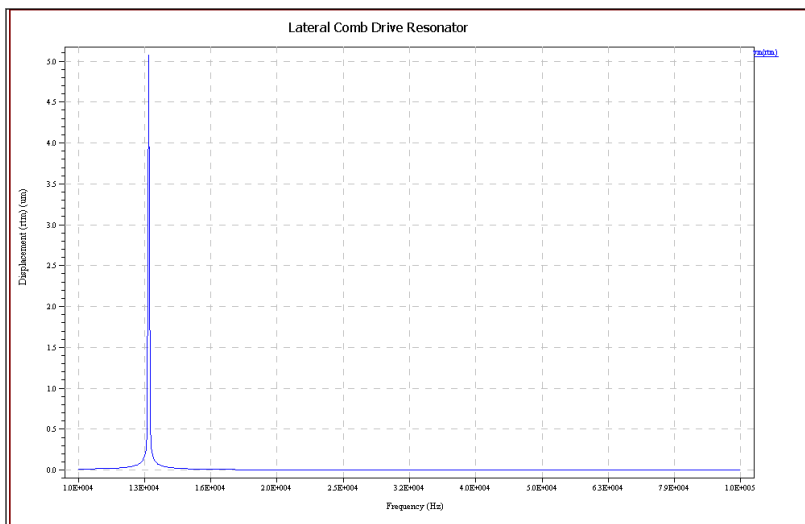


Figure 14: Customizing the amplitude wave form

You may want to measure specific values on the chart. Cursors are vertical lines, horizontal lines, or + points that can be used to identify locations on the trace for measuring. The vertical and horizontal line cursors extend to the full height or length of the chart. As the line cursors are moved around the chart, their

coordinates appear at the margins of the chart window. We demonstrate the use of **Vertical Bar** cursors below.

- ☑ Select **Chart > Cursors > Vertical Bars**.

Two vertical cursors should appear on the chart. They can be dragged with the left mouse button. Their **X** axis locations (**x1** and **x2**) and the difference in X axis locations (**dx**) are displayed on the top left corner of the chart window. The **Y** axis location of the moving or last moved cursor is displayed under the trace name on the right side of the chart.

- ☑ Position the left bar so that it lines up with the tip of the trace peak.

- ☑ Select **View > Mouse Zoom**. Click and drag a box around the tip of the peak in **vm(rtm)**. Make sure your drag box is within the chart window.

W-Edit will change the magnification of the chart so that the area outlined by the box fills an entire window. The smaller the box is drawn, the closer W-Edit will zoom into the chart. Continue magnifying your view until you have a clear view of the peak.

If you have zoomed-in too closely, you can retreat. Be sure that your mouse is in the window you want to adjust, then select **View > Zoom Out**.

- ☑ Position the left cursor so that it lines up with the tip of the trace peak.

The frequency value is shown as **x1** and should be about 13 kHz. The displacement value can be found under the trace name, **vm(rtm)**, and should be about 5.1  $\mu\text{m}$ .

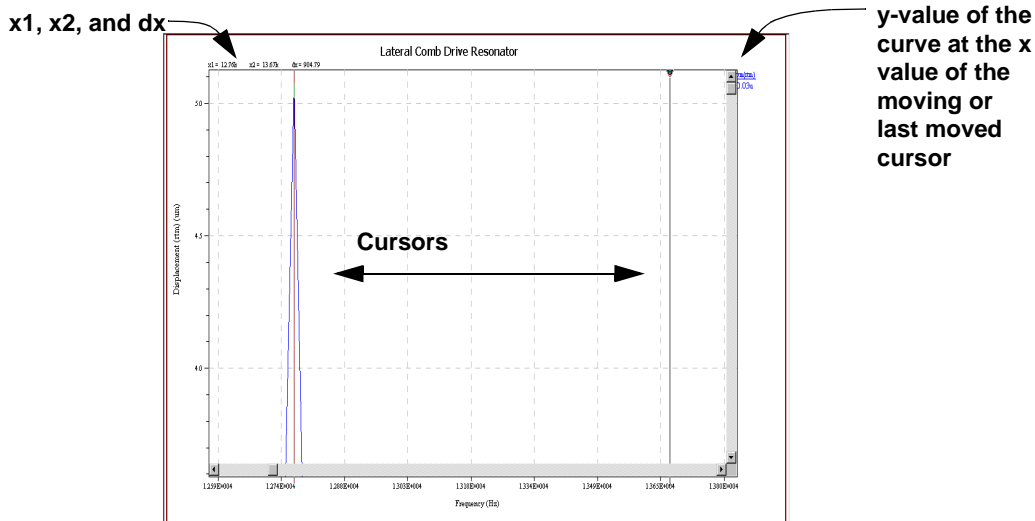


Figure 15: Using the **Vertical Bars** cursor



## Generating a Layout


- ☑ You may now exit S-Edit, T-Spice and W-Edit, if you like, using **File > Exit**.

Now, you will learn how to create, from MEMS layout components, the plate, comb-drives and springs used in the resonator.

### Tutorial Breakpoint

You will now generate the layout of a resonator using the **MEMS Layout Palette**.

### Launching L-Edit

- ☑ Launch L-Edit by double-clicking the **L-Edit** icon  in the installation directory. A default file named **Layout1** should be visible in the work area.

The L-Edit user interface is similar in appearance to the S-Edit user interface. In addition to the menu, palette, and status bars, there is a *shortcut bar* that contains buttons for the most commonly-used menu commands.

As in S-Edit, L-Edit files are assembled hierarchically from discrete, usually functionally distinct, units called *cells*, which can be edited and instantiated.

The *current* file and cell are named at the top of the application window.

## Opening the File

- ☑ Select **File > Open** to open the *reson.tdb* file. The **Resonator** cell should appear as the active cell. Use this view of the **Resonator** cell of the *reson.tdb* file as a reference while you work through this section.
- ☑ Make the **Layout1** file active by selecting the window containing **Cell0** of **Layout1** from the list of windows under the **Window** menu.

## Creating Components

The mask layout for MEMS components can be created using the **Library Palette** accessed via the **Library** option of the MEMS Pro Toolbar.

The **MEMS Library Palette** contains active elements, passive elements, test elements and resonator elements. The resonator element collection contains all the parts you will need to create a resonator. All of these parts can be created manually, using the drawing primitives available in MEMS Pro, but that task would be tedious and time consuming.

The **MEMS Library Palette** should have been loaded as part of your installation setup.

- ☑ Check that the **MEMS Pro Toolbar** has automatically appeared in the L-Edit window.

If the **MEMS Pro Toolbar** is not automatically loaded, you will need to load it manually. Refer to the introduction of Chapter 3 - MEMS Pro Toolbar.

### *Using the MEMS Library Palette*

- ☑ Select **Library > Library Palette** in the **MEMS Pro Toolbar** to invoke the **Library Palette** dialog box.

The **MEMS Library Palette** dialog box contains four tabs: **Active Elements**, **Passive Elements**, **Test Elements**, and **Resonator Elements**. You will use the **Resonator Elements** in this tutorial.

- ☒ Select the **Resonator Elements** tab to make the resonator components available.

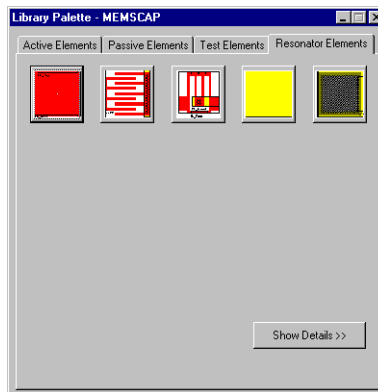



Figure 16: **MEMS Layout Palette**

### *Generating the Plate*

- ☒ Click the **Plate**  button to invoke the plate generation macro.  
A dialog box will appear requesting the parameters of the plate.

- ☒ Enter **100** as the **Width** and click **OK**, accepting the default values for the other parameters.

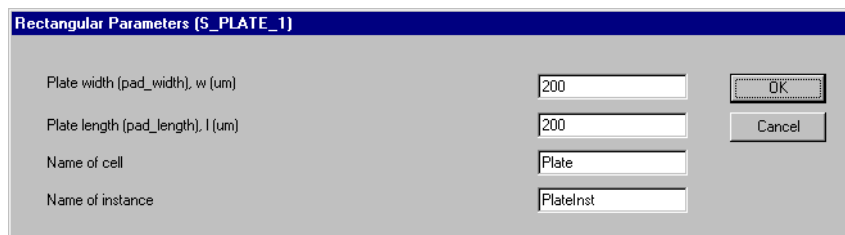


Figure 17: **Plate Parameters** dialog

L-Edit/UPI now creates a plate matching the input parameters. To see the entire plate, home the view by selecting **View > Home** or by pressing the **Home** key. The plate shown in the active window is an instance (**PlateInst**) of a newly created cell named **Plate**.

Each cell name must be unique in a file. Therefore, you should set the **Name of Plate Cell** to something other than **Plate** when running the macro again in the same file.

### *Generating the Comb-drives*

- ☒ Click the **comb-drive**  button to create a lateral comb-drive.

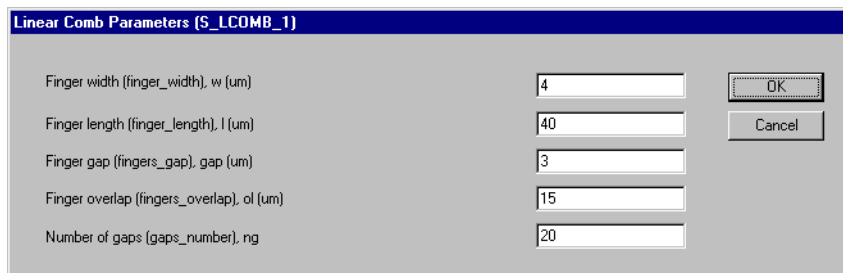
- ☒ Change the **Name of Instance** to **CombRight** and click **OK**.
- ☒ Once a comb-drive appears on the screen, zoom out by pressing the minus key several times or by selecting **View > Zoom Out**.

### *Editing an Already Generated Component*

You will now learn to edit a component once you have created it. The component you will edit is the comb-drive you have just instantiated.

- ☒ Select the comb drive
- ☒ Select **Library > Edit Component**.

The **Linear Comb Parameters** dialog box appears.



The dialog box titled "Linear Comb Parameters (S\_LCOMB\_1)" contains five input fields and two buttons. The input fields are labeled with their respective parameters and units: Finger width (finger\_width), w (um) with value 4; Finger length (finger\_length), l (um) with value 40; Finger gap (fingers\_gap), gap (um) with value 3; Finger overlap (fingers\_overlap), ol (um) with value 15; and Number of gaps (gaps\_number), ng with value 20. The buttons are labeled "OK" and "Cancel".

Parameter	Value
Finger width (finger_width), w (um)	4
Finger length (finger_length), l (um)	40
Finger gap (fingers_gap), gap (um)	3
Finger overlap (fingers_overlap), ol (um)	15
Number of gaps (gaps_number), ng	20

Figure 18: **Linear Comb Parameters** dialog box

- ☒ Set the **Number of gaps** to **21** and click **OK**.

The modified comb drive appears in the L-Edit window.

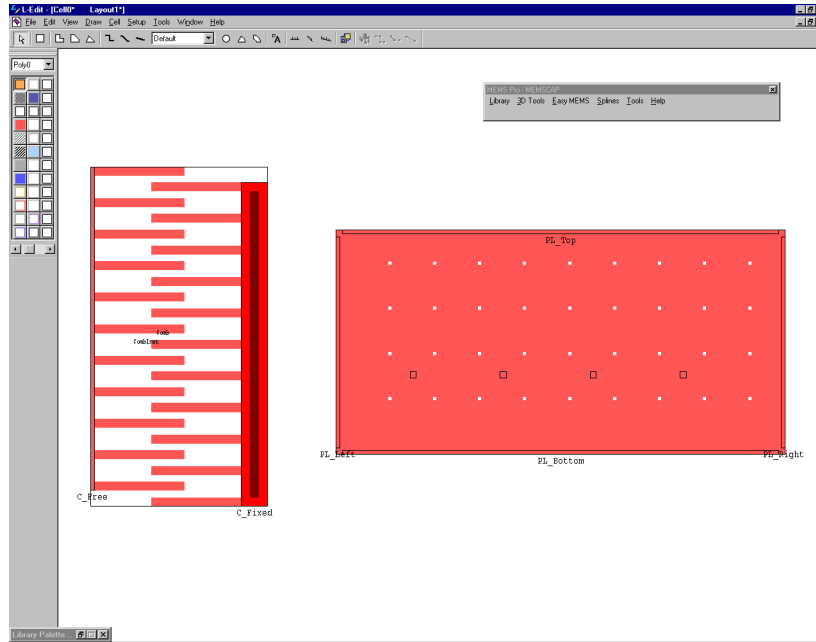


Figure 19: Viewing the modified comb-drive

This editing possibility is quite useful. Indeed, if you instantiate a component and then decide that it should be larger or longer, you can just modify one of its parameters and the newly edited component is automatically instantiated.

### *Attaching Components*

- ☑ Drag the comb-drive to the right side of the plate so that the two objects slightly overlap.

Recall that an object can be dragged to new locations by selecting it with a click, and then holding down the center mouse button while moving it to the desired location on the page. For two-button mice, left-click on the object while holding down the **Alt** key to accomplish the move.

- ☑ Zoom in with the plus key. Use the arrow keys to pan the view to where the comb-drive overlaps the plate.



- ☑ Re-align the **comb-drive** so that it looks like the figure below.

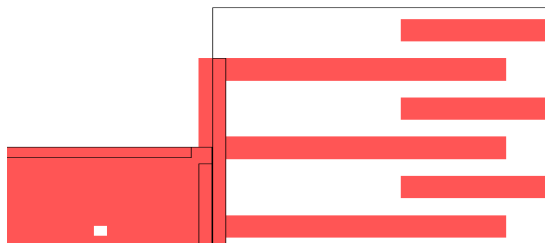


Figure 20: Aligning the comb-drive to the plate

- ☑ Copy the comb-drive by clicking it and choosing **Edit > Copy**, then **Edit > Paste**. The new comb-drive will appear in the center of the page, on top of your existing drawing. Move it to the side of the other layout objects.
- ☑ Flip the second comb-drive by selecting **Draw > Flip > Horizontal**.
- ☑ Change the name of this copied instance by selecting **Edit > Edit Object** and entering **CombLeft** in the **Instance Name** field.
- ☑ Attach the second comb-drive to the left side of the plate (Figure 21).

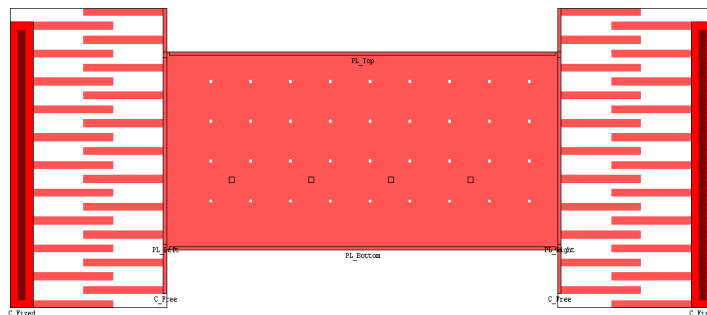



Figure 21: Viewing the uncomplete design

### *Generating the Folded Springs*

- ☑ Create a folded spring by clicking the **Folded Spring** button from the **Library Palette**. 
- ☑ Change the **Name of Instance** to **SpringTop** and click **OK**.

- ☒ Position it above the center of the plate so that it overlaps (see Figure 22).

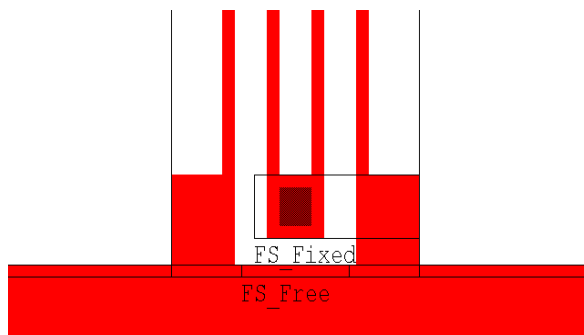



Figure 22: Positioning the folded spring


- ☒ Copy and paste ***SpringTop***. Then, select **Edit > Flip > Vertical**. Position the new folded spring below the plate.
- ☒ Change the name of this copied instance by selecting **Edit > Edit Object** and entering ***SpringBottom*** in the **Instance Name** field.

### *Generating the Ground Plate*

- ☒ Create a ground plate by clicking the **Ground Plate**  button from the **Library Palette**. Leave all the parameters at their default values and click **OK**.


- ☑ Home the view, then move the ground plate so that it covers all the moving parts of the resonator (refer to Figure 24).

### *Generating the Bonding Pads*

- ☑ Create a bonding pad by clicking the **Bonding Pad**  button. Leave all parameters for the bonding pad at their default values. Click **OK**.

- ☑ Position the bonding pad slightly to the right of the right comb-drive (Figure 23).

Now, you must connect the bonding pad to the comb-drive by drawing a box on **Poly0** overlapping the two components.

- ☑ Choose the **Box** tool  by clicking it and select the **Poly0** layer from the **Layers Palette** by clicking on the first item in the first row of the **Layers Palette**. As your mouse is moved over the **Poly0** button, a tool tip will appear displaying the layer name. **Poly0** will also appear in the list box at the top of the **Layers Palette**.

- ☑ Click once to set the upper left corner, hold the key down and drag to the opposite corner, and release.

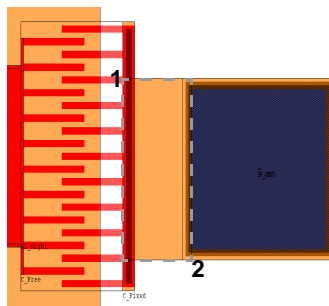


Figure 23: Attaching the first bonding pad

- ☑ Copy the bonding pad and place it to the left of the left comb-drive.
- ☑ Flip the bonding pad by selecting **Draw > Flip > Horizontal**.
- ☑ Connect the bonding pad to the comb-drive by drawing a box on **Poly0** overlapping the comb-drive and the bonding pad in a similar fashion as above.
- ☑ Make a third copy of the bonding pad and place it to the bottom left side of the ground plate.

- ☑ Connect the ground plate to the bonding pad by drawing a box on **Poly0** overlapping the bonding pad and the ground plate in a similar fashion as above.
- ☑ Change the name of the cell you have been working on from **Cell0**. Select **Cell > Rename**. Enter **MyResonator** as the cell name.
- ☑ Save the file by choosing **File > Save**. Enter **myreson.tdb** as the file name and click **OK**.

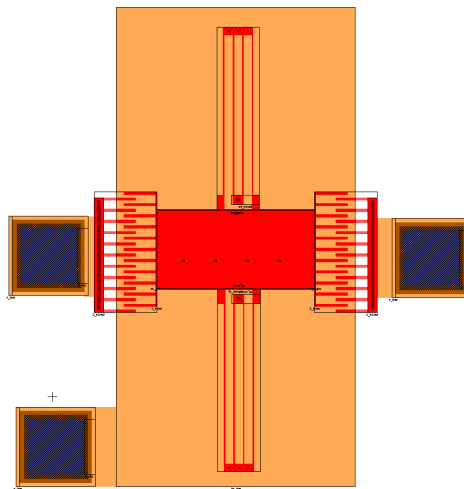


Figure 24: Final view of the lateral resonator

If you are interested in performing layout netlist extraction and layout vs. schematic comparison, refer to Chapter 11 - Verification and use the **myreson.tdb** file.

## Viewing Properties

Properties can be attached to any L-Edit object including boxes, polygons, wires, circles, ports, rulers, instances, cells, and files. Properties can contain supplementary but necessary information about an object, such as what color it will appear when modeled, or what its constituent material is.

MEMS Pro library components have a properties category called **Extract Properties**. This category provides a link between a design layout and its netlist description. **Extract Properties** are accessed via the **Cell Info** dialog box or the **Edit Instance** dialog box.

### Note

---

For more information on *properties and extraction*, see Properties on page 1-66 and Extracting Layout on page 3-48 of the *L-Edit User Guide*.

---



Select the instance of the **Plate** and choose **Edit > Edit Object**.

- ☑ Click the **Properties** button. This instance should have no properties attached to it. If the instance does not have **Extract Properties**, L-Edit/Extract pursues the hierarchy and looks for extract properties on the parent cell.
- ☑ Select **Cell** from the **Parent** list box and click the **View Parent** button to view the properties for the **Plate** cell.
- ☑ To view the extract properties, click the **+** sign next to the **EXTRACT** folder.

Three properties should be displayed under the **EXTRACT** folder (Figure 25).

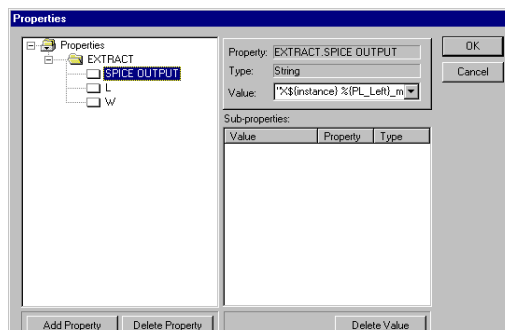


Figure 25: **Properties** dialog box



Try selecting different properties to view their types and values on the right side of the display.

- ☑ Click the **L** property. The value that you entered to represent the plate length is shown as the **Value** on the right side of the dialog.

To return to the layout, perform the following operations:

- ☑ Click **Cancel** to exit this **Properties** dialog box.
- ☑ Click **Cancel** to exit the main **Properties** dialog box.
- ☑ Click **Cancel** to exit the **Edit Object(s)** dialog box.


## Viewing a 3D Model

The 3D Model Viewer automatically generates a 3D model from a layout and a process definition.

### Tutorial Breakpoint

You will now create and view a solid model. If you are beginning the tutorial now, follow the next section on Launching L-Edit and Opening a File to open the design file you have been provided with. If you are continuing from the previous section, you may use your own design and skip to Process Definition on page 74.

### Launching L-Edit and Opening a File

- ☑ Launch L-Edit by double-clicking the **L-Edit** icon  in the installation directory. A default file named **Layout1** should be visible in the work area.
- ☑ Close the **Layout1** file by selecting **File > Close**.

- ☒ Using the **File > Open** command, open the **reson.tdb** file (Figure 26) .

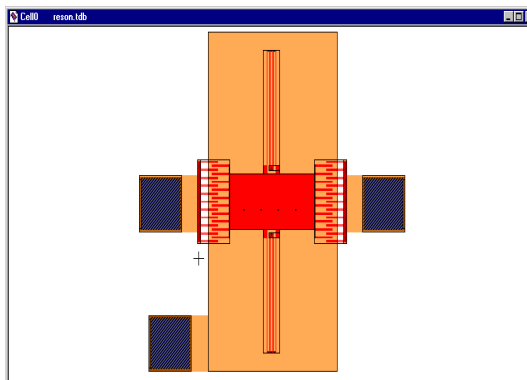


Figure 26: Layout view of the resonator

## Process Definition

### *Importing the Process Definition*

In addition to layout or mask data, the 3D Modeler needs the fabrication process description to generate 3D models. This information may already be saved with the layout, if not, it must be imported into the design file.

The following setup procedure loads the process information into the design file. Once imported, the process definition information is saved with the design file; it need not be re-imported when the file is re-opened.

- ☑ Choose **3D Tools > Edit Process Definition** in the MEMS Pro Palette. In the **Process Definition** dialog, click the **Import** button.
- ☑ In the **Open** dialog box, select **mumps\_i.pdt** and click the **Open** button.

**Note**

---

For more information on *process definitions*, see Process Definition on page 352 of the *MEMS Pro User Guide*.

---

Information describing the MCNC MUMPs process is imported into the dialog box. This information, in conjunction with the open layout, is used to build a 3D model.

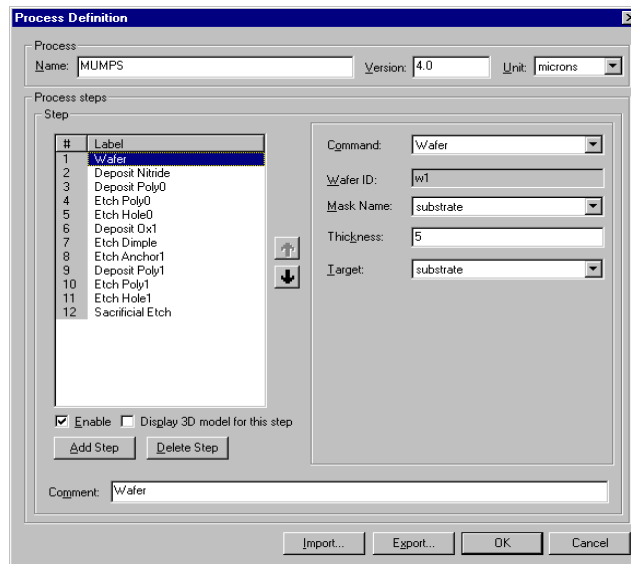


Figure 27: **Process Definition** dialog box

The top of the dialog box contains identifying information for the process definition. The left side of the dialog contains a list of the process steps. Beneath

it and to its side are the controls for adding, deleting, moving, enabling, and displaying 3D models for intermediate steps. The right side of the dialog contains the parameters of the selected step in the process steps list.

### Note

---

For more information on *editing process definitions*, see Editing the Process Definition on page 149 and Process Steps on page 357 of the *MEMS Pro User Guide*.

---

**Mumps\_i.pdt** is an abridged version of the MUMPS process definition. It includes the steps up to the patterning of the second polysilicon layer. (A simplified process is used for the tutorial in the interest of saving computation time.) You do not lose any important information because the linear resonator structure is defined by the process steps up to the patterning of the second polysilicon layer. Note though, that the bonding pads are composed of the stacking of the first polysilicon layer, third polysilicon layer, and the metal layer, so the 3D representation of the bonding pads will be incomplete.

- ☒ To attach the MUMPS process information to the design database, click **OK** to close the **Process Definition** dialog.

## 3D Model View

### *Generating the 3D Model*

- ☑ Click somewhere in the title bar of the resonator layout window to make it active.
- ☑ Choose **3D Tools > View 3D Model** in the MEMS Pro Palette.

The 3D model generation will begin and a progress dialog will appear. In a few minutes, the 3D model will appear in a new, active, L-Edit window. When a **3D Model View** window is active, the menu bar changes and the **3D Model View** toolbar buttons become enabled.


### *Manipulating the 3D Model View*

To manipulate the 3D model view, use either the **3D Model View** toolbar or the menu options under the **View** menu.




Figure 28: **3D Model View** Toolbar


The **Orbit View** allows you to rotate the model in order to view it from any angle.

- ☑ Select **View > Orbit** or click the **Orbit**  toolbar button.
- ☑ Click and drag over the 3D model window to orbit the model.

The 3D model view may be translated by panning the view.

- ☑ Select **View > Pan** or click the **Pan**  toolbar button.
- ☑ Click and drag over the 3D model window to pan.

You may examine the details of the model by zooming in to the area of interest.

- ☑ Select **View > Zoom > Box** or click the **Window Zoom**  toolbar button.
- ☑ Click and drag the pointer to the opposite corner of the zoom box and release the mouse button.

You may also use the **Ctrl** key and the three mouse buttons to **Orbit**, **Pan**, and **Zoom**:

- **Ctrl+Left** click and drag over the 3D model window to **Orbit**.
- **Ctrl+Right** click and drag over the 3D model window to **Zoom** in and out.
- **Ctrl+Center** click and drag over the 3D model to **Pan** the view.



**Note**

---

For more information on *changing your point of view of the solid model*, see Accessing 3D Models on page 110 of the *MEMS Pro User Guide*.

---

### *Multiple Views*

Multiple views of the generated 3D model may be viewed simultaneously.

- ☒ Click somewhere in the title bar of the layout window to make it active.
- ☒ In the MEMS Pro Palette, select **3D Tools > View 3D Model** twice to create two more views of the 3D model.

- ☑ Select **Window > Tile** to tile the windows. All open windows will be resized so that they fit without overlapping (Figure 29). Each 3D model view may be manipulated independently.

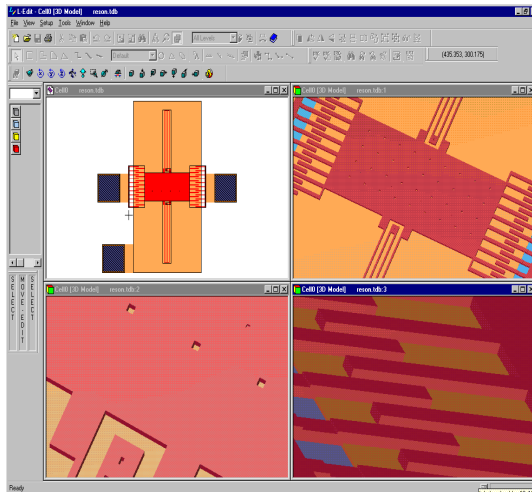



Figure 29: Tiling the windows displaying the 3D generation steps


## *Viewing the 3D Model*

Once generated, 3D models do not need to be regenerated to be viewed again. In addition, 3D models are saved with the design information into the Tanner Database **.tdb** file.

- ☑ Keep the layout window with the original design open and active; close all 3D model windows by clicking the  button in the upper right corner of each window.
- ☑ Select **3D Tools > View 3D Model** in the MEMS Pro Palette and the 3D model will reopen without generating.

## 3D Cross-section

Cross-sections may be taken from the 3D model using the **Cross-section** tool.

- ☑ Click somewhere in the title bar of the layout window to make it active.
- ☑ Select **3D Tools > View 3D Model** in the MEMS Pro Palette to create another view of the 3D model.
- ☑ Click the **Cross-section** tool . The **Generate 3D Model Cross-Section** dialog will appear (Figure 30).

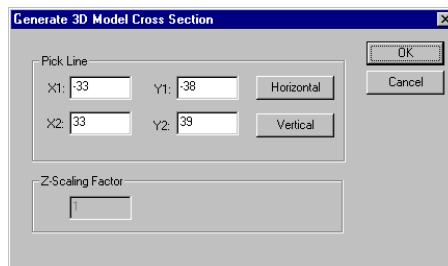


Figure 30: **Generate 3D Model Cross Section** dialog

- ☑ Click **OK**. A new L-Edit window will appear with a cross-section view.

The **3D Model View** window will snap to the top view and a line representing the cross-section cut plane will be displayed on top of the model. The cross-section plane is always normal to the surface of the wafer.

- ☑ Select **Window > Tile** so that you can view all the open windows at once.

- ☑ Manipulate the cross-section plane line to the desired location by using the left mouse button to move the end points of the line. The cross-section window will be updated with each manipulation of the line.

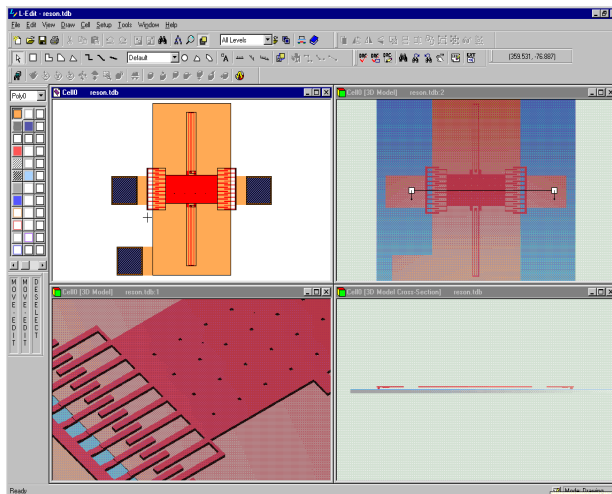


Figure 31: Tiling the windows displaying the various cross-section steps

- ☑ To exit the cross-section mode, select a different mode by clicking a toolbar button or by selecting the menu item corresponding to the next desired function.

**Note**

---

For more information on cross-sections, see [Viewing a Cross-section](#) on page 141.

---





# Drawing Tools

In this section of the tutorial, we will explore the drawing tools available with MEMS Pro. MEMS Pro supports objects such as all-angle wires and polygons, arcs, tori, circles, splines, and curved polygons. We will use some of these to draw a rotary side-drive motor.

A special tutorial to use splines is given in Chapter 4 - Splines.

## Tutorial Breakpoint

- 
- ☑ If you are starting the tutorial here, double-click the **L-Edit** icon.
- 

Ten object types are supported:

- Box
- Polygon
- Wire
- Circle
- Arc
- Torus
- Splines

- Port
- Ruler
- Instance

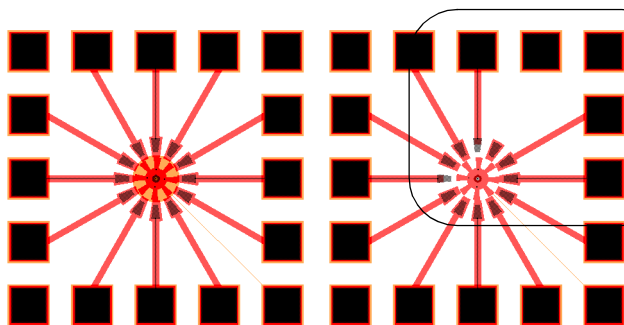
**Note**

For more information on *drawing objects*, see Drawing and Editing Objects on page 1-240 of the *L-Edit User Guide*.



From L-Edit, open the **motor.tdb** file.

complete  
motor  
example



Area to be  
completed

In the visible cell, **Demo**, there are complete (left) and incomplete (right) layouts of a rotary side drive electrostatic motor. This part of the tutorial will guide you in finishing the incomplete design.



## Drawing a Wire

On the incomplete motor design, a pad is not attached to a stator on the *poly1* layer. A wire must be drawn to connect this pad to its stator.

The anchor point is the first vertex of a wire. Wires can have several vertices.

- ☑ Select **View > Zoom > Mouse** so that the pad and torus are visible as in Figure 32. Left-click at one box corner, hold the button down as you drag to the opposite box corner, and release.

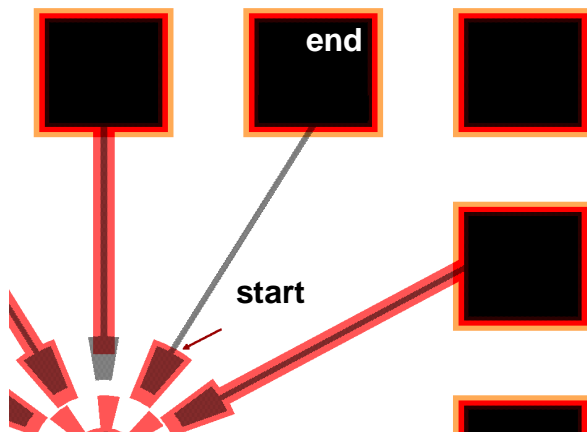



Figure 32: Wiring the pad to the stator

- ☑ Select **All Angle Wire** by clicking the  button in the **Drawing** toolbar.

- ☑ Choose the **poly1** layer from the **Layer Palette**.

The mask layers are displayed in the **Layer Palette** as an arrangement of square icons that represent the available layers. The icons are differentiated by color and pattern. As you move the cursor over an icon, the name of the layer beneath the cursor appears in the **Status** bar. A layer is selected by clicking the corresponding icon.


- ☑ Click the stator opposite the pad to start drawing a wire. Successive clicks will produce intermediate points of connection. Right-click the pad to end the wire.

It is important that the wire touch the **poly1** layer of the torus and the pad or these elements will not be connected. When the drawing operation is completed, that new object remains selected. You will now change the width of the wire.

- ☑ Select **Edit > Edit Object**. Change the **Wire Width** to **15** locator units and click **OK**.

## Drawing a Torus

When drawing a torus, the first click of the left mouse button sets the center. The second click determines the inner radius of the torus. The third click decides the outer radius and the sweep angle.

- ☑ Select the **Torus** tool  from the **Drawing** toolbar.
- ☑ Left-click the center of the incomplete motor to begin drawing the torus. Left-click at the inner radius point, then complete the torus by right-clicking at the outer radius point (Figure 33).

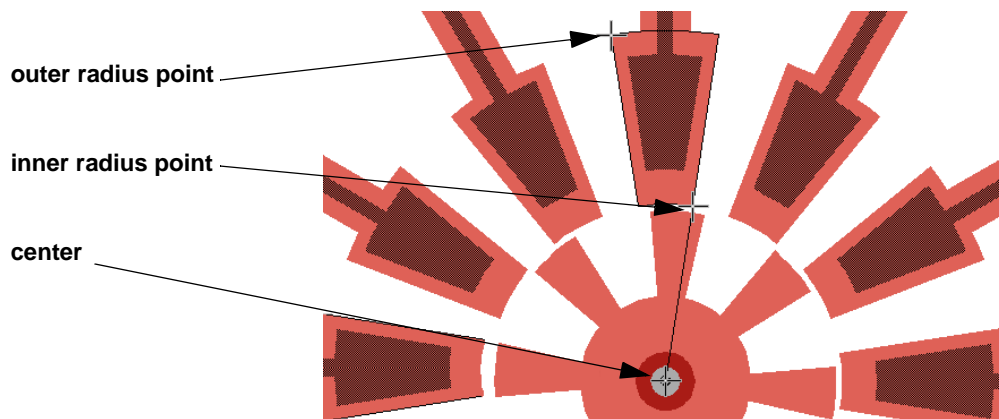



Figure 33: Creating a torus

## Drawing a Curved Polygon

To create a curved polygon, a straight edged polygon must be drawn first. The straight edges can then be converted to curved edges by selecting a given edge with a **Ctrl+Right** click and then dragging out the desired curve with the center mouse button.

You will use a curved polygon to draw a stator like the one you drew with the torus tool.

- ☒ Select the **Window Zoom** tool to arrange the view in such a way that the left-most stator is visible (Figure 34).
- ☒ Select the **All Angle Polygon** tool  from the **Drawing** toolbar.

- ☑ Left-click the first numbered vertex to begin drawing the polygon. Left-click the second and the third vertices, then complete the polygon by right-clicking the fourth numbered vertex (Figure 34).

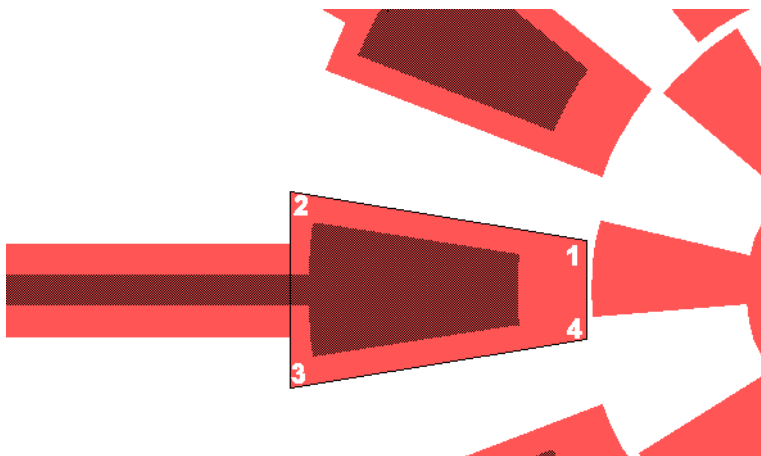


Figure 34: Creating the all angle polygon

- ☑ Select the rightmost edge with a **Ctrl+Right** click. Once selected, the edge will be highlighted.

- ☑ Press the **Ctrl** key, hold and drag the center mouse button (**Alt+Left** hold for two button mice) to the left to convert the straight edge into a curved edge as shown below. Release the mouse button to complete this action.

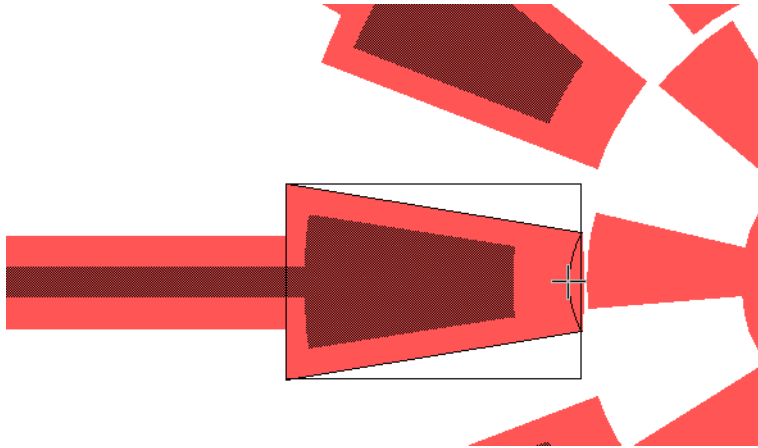


Figure 35: Curving the rightmost edge of the polygon

- ☑ Similarly, select the left edge with a **Ctrl+Right** click. Once selected, the edge will be highlighted.

- ☑ Press the **Ctrl** key, hold and drag the center mouse button (**Alt+Left** hold for two button mice) to the left to convert the straight edge into a curved edge. Release the mouse button to complete this action (Figure 36). .

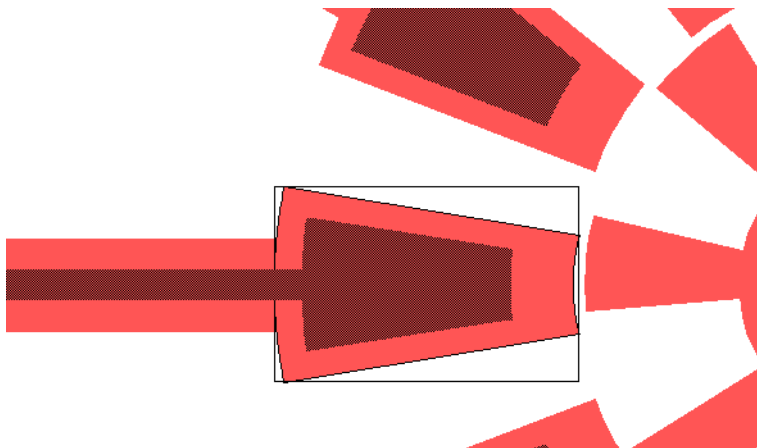


Figure 36: Curving the leftmost edge of the polygon


---

**Note**

For more information on *drawing and editing curves*, see Curves on page 1-254 of the *L-Edit User Guide*.

---

## Drawing a Circle

- ☑ Select the **Circle** tool  and select the *poly0* layer.
- ☑ Place a circle at the center of the incomplete design so that it matches the completed design. Left-click to anchor the center of the circle; drag the mouse to set the radius of the circle and release.

## Drawing a Box

Close inspection of the rotor reveals there are three dimples positioned at 90° intervals near the center of the rotor. A fourth dimple must be placed on the *dimple* layer to complete the pattern.

- ☑ Choose the **Box** tool  and select the *dimple* layer.

The box may be constructed anywhere on the layer. We will move it to the proper location after it is complete.

- ☑ Left-click to anchor the first corner of the box, drag away from the anchor point to determine the opposite corner of the box three grid units away, and release.
- ☑ Left-click the newly drawn dimple box to select it.
- ☑ Move the dimple into place with a center-click (**Alt+Left** click for two button mice) and hold, drag and release at the desired location. The dimple should be



placed to the right side of the rotor so that it is approximately  $90^\circ$  from the next nearest dimple (Figure 37).

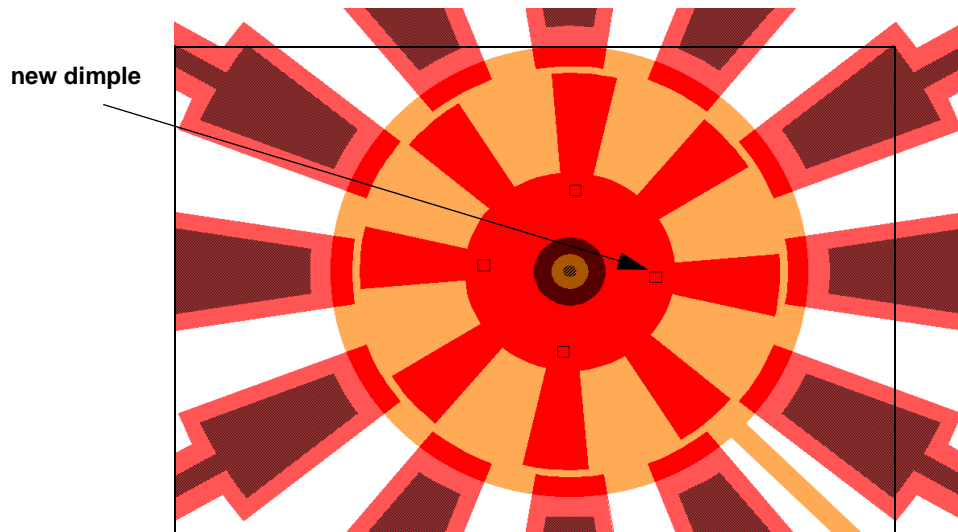


Figure 37: Creating a dimple

***This concludes the basic tutorial of MEMS Pro.***

# 3 MEMS Pro Toolbar

---

▪ Introduction	98
▪ Library Menu	103
▪ 3D Tools Menu	109
▪ Easy MEMS Menu	114
▪ Splines	118
▪ Tools	121
▪ Help	124

## Introduction

The new **MEMS Pro Toolbar** (see Figure 38) offers the opportunity to access relevant MEMS-specific features. These features are either former options that were previously accessible from the **Tools** menu of the L-Edit menu bar (Library Palette, **3D Tools** menu and **Polar Array** option) or new created features (**Plate Release** option, **Splines** menu, and **Tools** menu) that facilitate the designer's tasks.

The **MEMS Pro Toolbar** can be launched automatically at the opening of the L-Edit session. Its purpose is to better expose all the MEMS Pro features to the user.

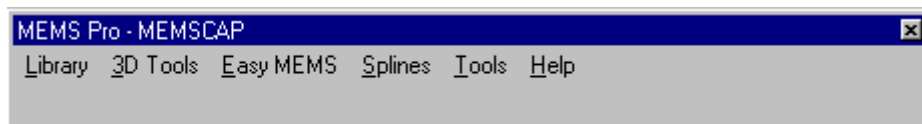


Figure 38: **MEMS Pro Toolbar**

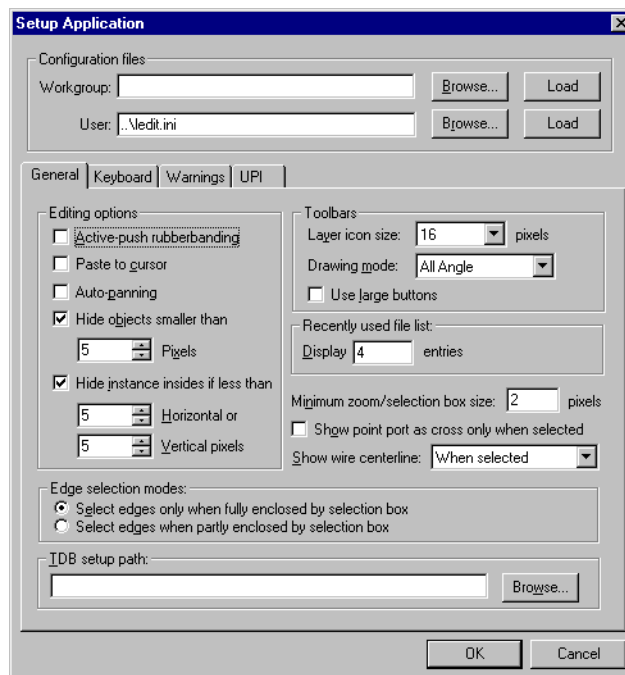
If the **MEMS Pro Toolbar** is not automatically loaded, perform the following operations:

- ☑ Select **Tools > Macro** to invoke the **Macro** dialog.
- ☑ Click the **Add** button to bring up the **Open** dialog. Select **MEMSPHysical.dll** in the **memslibs** directory.

The **MEMS Pro Toolbar** appears.

- ☑ Click the **Close** button to exit the **Macro** dialog.
- ☑ Then, select **Setup > Application**.

The **Setup Application** dialog box appears.

Figure 39: **Setup Application** dialog box

- ☑ Browse to the **ledit.ini** file located under **Program Files / Memscap / MEMS Pro v3.00** and click **OK**.
- ☑ Click **Tools > Macro**.

The **Macro** dialog box appears.

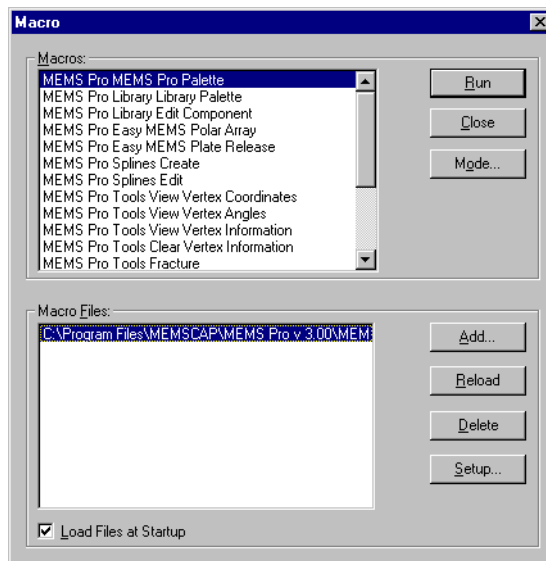


Figure 40: **Macro** dialog box

- ☒ Select the **MEMSPHysical.dll** in the bottom area and click the **Load at startup** check box.
- ☒ Click **Close**.

Your **MEMS Pro Toolbar** will now appear automatically.

## Library Menu

The **Library** menu offers two possibilities:

- Accessing the **Library Palette**
- Editing components created using the **Library Palette**

## Library Palette

The **Library Palette** contains a variety of components that can be assembled to create a full MEMS device. It allows the instantiation of active, passive and test elements but also resonator elements.

- ☑ Access the **Library Palette** (see Figure 41) by selecting **Library > Library Palette** in the **MEMS Pro Toolbar**.



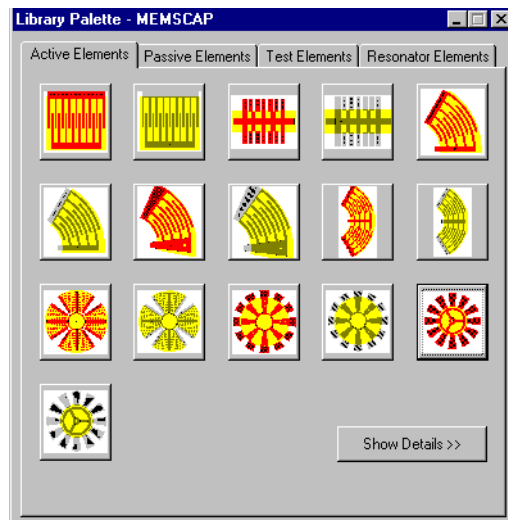


Figure 41: **Library Palette**

**Note**

For more information on *discovering and using the Library Palette*, refer to Chapter 15 - MEMSLib Reference.

## Edit Component

You have now the possibility to modify the components that you have previously created using the **Library Palette**.

To edit components, perform the following steps:

- ☒ Select the component you want to edit.

In this example, consider a harmonic side drive (refer to Figure 42).

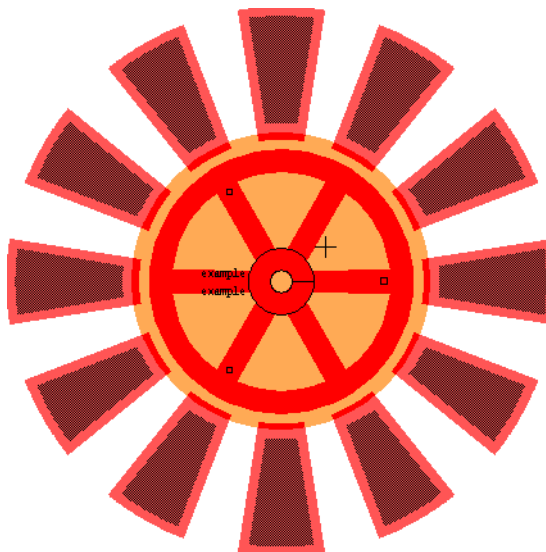
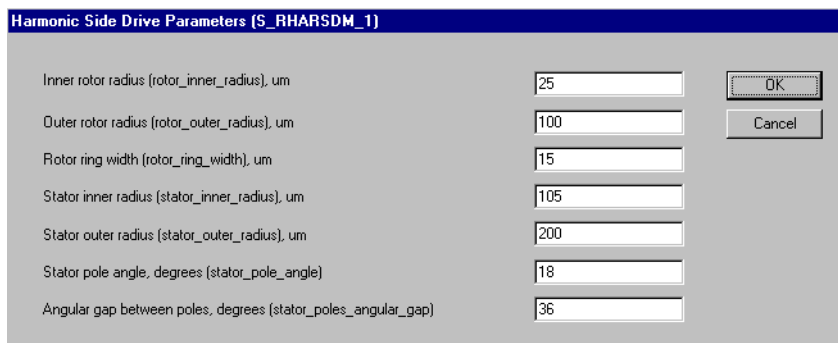


Figure 42: Default harmonic side drive

- ☒ Select **Library > Edit Component**.

A dialog box displaying the parameters of the selected component appears (Figure 43).



The image shows a software dialog box titled "Harmonic Side Drive Parameters [S\_RHARSDM\_1]". It contains six input fields for numerical values, each with a label and unit. The values are: 25, 100, 15, 105, 200, and 18. There are also two buttons, "OK" and "Cancel", on the right side of the dialog.

Parameter	Value
Inner rotor radius (rotor_inner_radius), um	25
Outer rotor radius (rotor_outer_radius), um	100
Rotor ring width (rotor_ring_width), um	15
Stator inner radius (stator_inner_radius), um	105
Stator outer radius (stator_outer_radius), um	200
Stator pole angle, degrees (stator_pole_angle)	18

Angular gap between poles, degrees (stator\_poles\_angular\_gap) 36

Buttons: OK, Cancel

Figure 43: Parameters dialog box of the harmonic side drive

- ☒ Modify the parameters values and click **OK**.

The component has been edited (Figure 44).

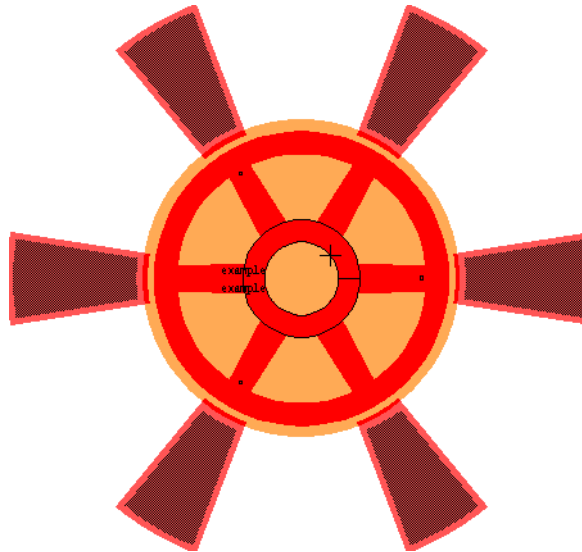


Figure 44: Edited harmonic side drive

## 3D Tools Menu

The **3D Tools** menu gathers all the options related to the 3D model generation and viewing. You can perform the following operations:

- Edit a process definition
- View a 3D model
- Delete a 3D model
- Export a 3D model

### Editing a Process Definition

- ☒ To edit a process definition, select **3D Tools > Edit Process Definition**.

The **Process Definition** dialog box appears allowing you to browse for the desired process definition file (.pdt) and to edit the various steps of the 3D generation (see Figure 45).

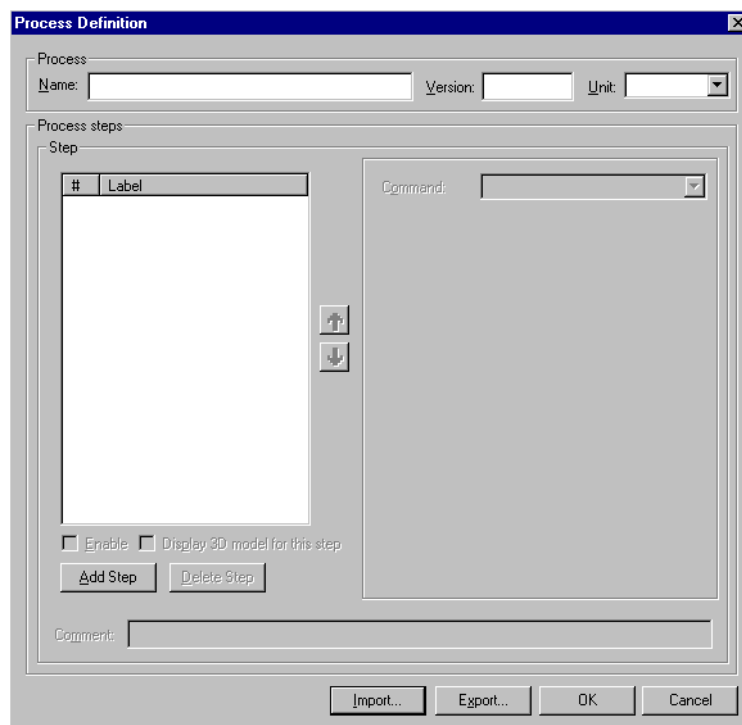


Figure 45: **Process Definition** dialog box

**Note**

---

For more information on *editing process definitions*, refer to Editing the Process Definition in Chapter 17 - Process Definition.

---

## Viewing a 3D Model

To view the 3D model of a layout, a process definition has to be previously determined (refer to Editing a Process Definition in the present document, for more information)

To view the 3D model of a layout, the layout has to be displayed in the L-Edit main window.



Then, select **3D Tools > View 3D Model** in the **MEMS Pro Toolbar**.

A progress bar called **Generating 3D Model** appears indicating which step is currently performed. At the end of the 3D generation, the 3D model appears in a new window.

**Note**

---

For more information on *viewing a 3D model*, refer to Viewing 3D Models from Layout of Chapter 6 - 3D Modeler.

---



## Deleting a 3D Model

To delete a 3D model, select **3D Tools > Delete 3D Model** in the **MEMS Pro Toolbar**.

The **Delete 3D Models** dialog box appears (Figure 46). You can choose to delete one cell, one file or all open files.

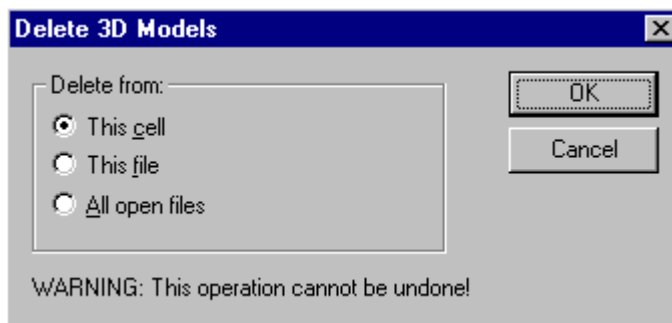


Figure 46: **Delete 3D Models** dialog box

### Note

For more information on *deleting 3D models*, refer to Deleting 3D Models of Chapter 6 - 3D Modeler.

## Exporting a 3D Model

To export a 3D model, select **3D Tools > Export 3D Model** in the **MEMS Pro Toolbar**.

The **Export 3D Model** dialog appears (Figure 47). You can export your 3D model into a .sat or a .anf file.

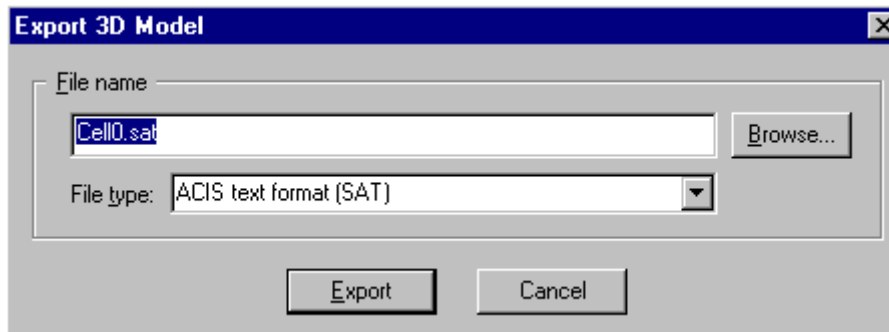


Figure 47: **Export 3D Model** dialog

### Note

---

For more information on *exporting 3D models*, refer to Exporting 3D Models of Chapter 6 - 3D Modeler.

---

## Easy MEMS Menu

The **Easy MEMS** menu offers two useful features that allow you to perform the following operations:

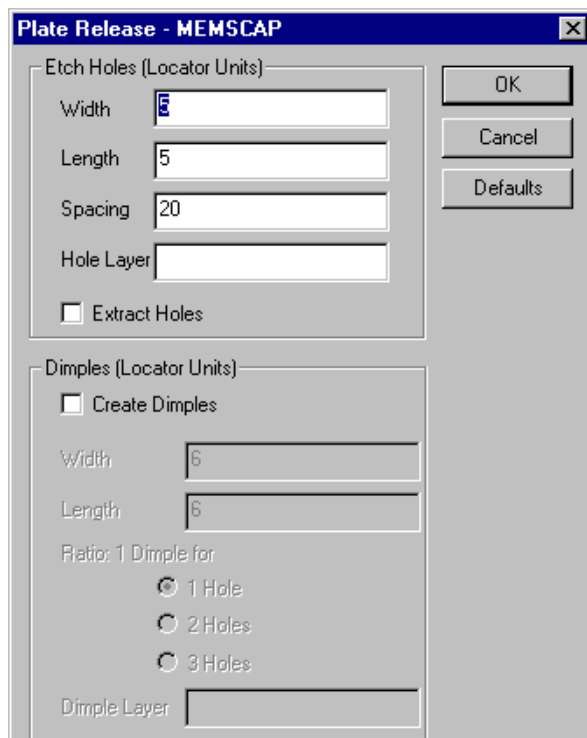
- Creating holes in a plate
- Customizing the duplication of elements

### Creating holes in a plate

To create holes in a plate, perform the following operations:

- ☒ On your layout, select the plate in which you want to create holes.
- ☒ Choose **Easy MEMS > Plate Release** in the **MEMS Pro Toolbar**.

The **Plate Release** dialog appears (Figure 48). You can define the width, length and spacing of the holes that will be created. You can also determine whether you want to create dimples or not. And, you can define not only the width and length of the dimples but also the ratio of dimples per hole.

Figure 48: **Plate Release** dialog

**Note**

---

For more information on *creating holes in a plate*, refer to *Generating Holes in a Plate* in Chapter 5 - MEMS Pro Utilities.

---

## Copying objects

The **Polar Array** feature allows you to copy objects around a reference point and to keep a regular angle between each object.



To use the **Polar Array** feature, select the element you want to duplicate and choose **Easy MEMS > Polar Array** in the **MEMS Pro Toolbar**.

The **Polar Array** dialog box appears (Figure 49). This feature depends on three parameters:

- The number of copies you want to create
- The angle for the copies
- The center of the array

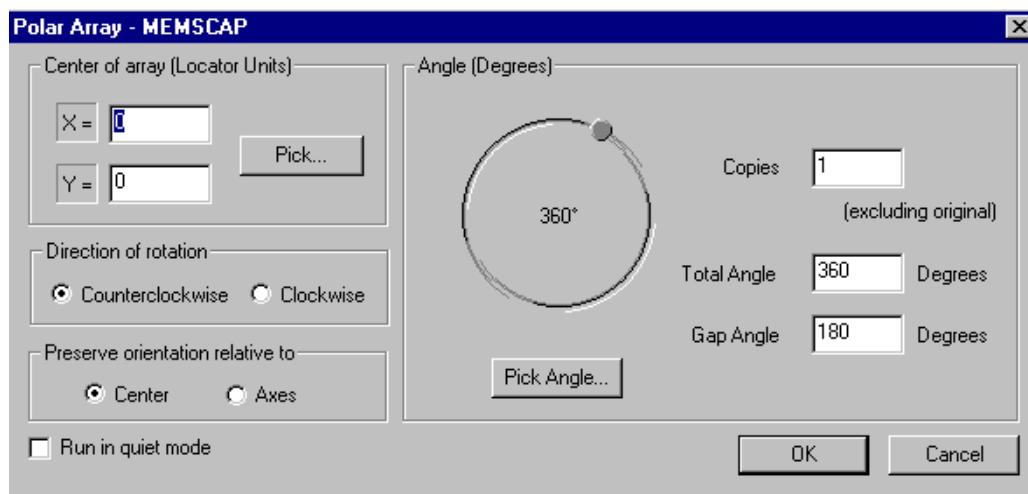


Figure 49: **Polar Array** dialog

**Note**

For more information on *using the Polar Array function*, refer to Generating Polar Arrays in Chapter 5 - MEMS Pro Utilities.

# Splines

The **Splines** feature is a new feature of MEMS Pro Version 3.0. It consists of the possibility of creating and editing splines.

## Note

---

For more information on the *creation and edition of splines*, refer to Chapter 4 - Splines.

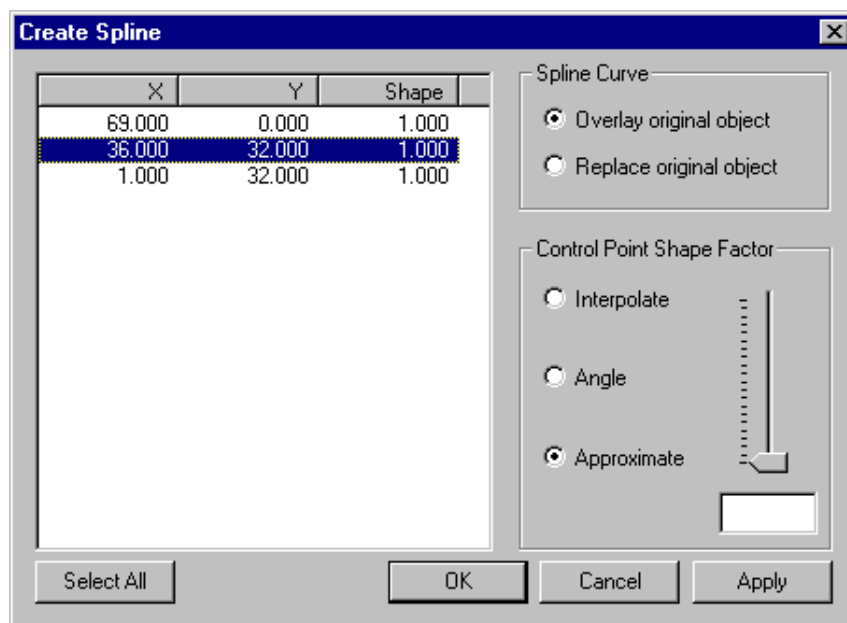
---

## Creating Splines

To create splines, perform the following operations:

- ☒ Select the reference wire or the object for which you would like to create spline-edge.
- ☒ Select **Splines > Create** in the **MEMS Pro Toolbar**.

The **Create Splines** dialog box appears (Figure 50). You have the possibility to extrapolate or approximate the reference object.

Figure 50: **Create Spline** dialog box



## Editing Splines

To edit a spline, perform the following steps:

- ☒ Select the spline you want to edit.
- ☒ Select **Splines > Edit** in the **MEMS Pro Toolbar**.

The same **Create Spline** dialog box (see Figure 50), in which you can modify the operation of creating a spline, appears.

### Note

---

This option also allows you to undo the spline creation, that is to re-create the original reference wire.

---

# Tools

The **Tools** menu gives you access to options related to the vertex coordinates, angles and information. You now have the possibility to view the coordinates, angles and information related to a vertex.

## Note

---

For more information on how to use these features related to vertices, refer to Viewing Vertex Coordinates and Angles in Chapter 5 - MEMS Pro Utilities.

---



## Viewing Vertex Coordinates



The **View Vertex Coordinates** feature of MEMS Pro V3.0 allows you to view the coordinates of selected elements.

To view the coordinates of a particular element, perform the following steps:

- ☒ Select the desired element.
- ☒ Choose **Tools > View Vertex Coordinates** in the MEMS Pro **Toolbar**.

The vertex number and coordinates of the selected element are displayed in the layout window.

## Viewing Vertex Angles

The **View Vertex Angles** feature allows you to view the angle values of selected elements.

To view the angle values of a particular element, perform the following operations:

- ☒ Select the desired element.
- ☒ Choose **Tools > View Vertex Coordinates** in the **MEMS Pro Toolbar**.

The vertex number and the angle values of the selected element are displayed in the layout window.

## Viewing Vertex Information

The **View Vertex Information** feature allows you to view the coordinates, the angle values and the number of the vertices of a selected element.

- ☒ To access that feature, select the element for which you want to view the information.
- ☒ Select **Tools > View Vertex Information** in the **MEMS Pro Toolbar**.

The information concerning the vertices of the selected element are displayed.

## Clearing Vertex Information

Once you have viewed the vertex information of a selected element, you can remove them from the design by selecting **Tools > Clear Vertex Information**.

# Help

The **Help** menu gives access to the MEMS Pro User Guide and to various information concerning MEMS Pro.

## MEMS Pro User Guide

To access the MEMS Pro User Guide, select **Help > MEMS Pro User Guide**.

## About MEMS Pro

To access information on MEMS Pro and its current version, select **Help > About MEMS Pro**.

The **About MEMS Pro** dialog box appears (Figure 51).

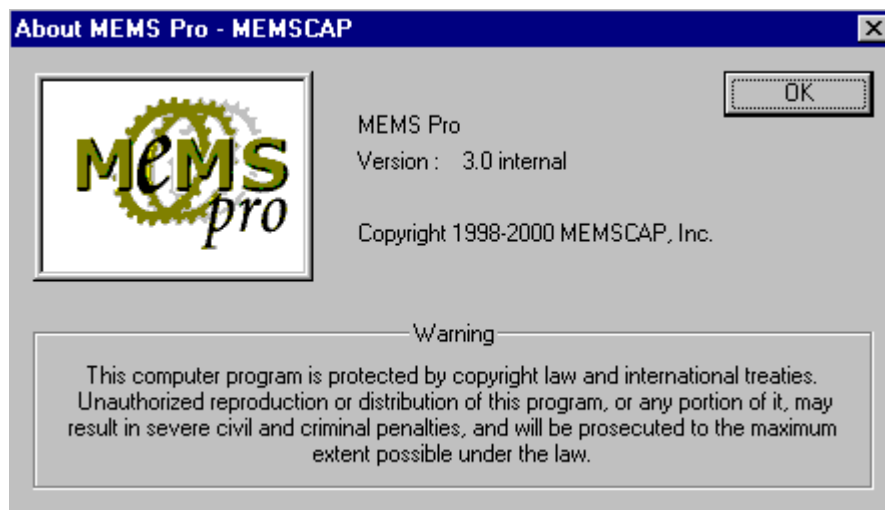


Figure 51: **About MEMS Pro** dialog box

# 4 Splines

---

▪ Introduction	127
▪ Create Spline Dialog Box	129
▪ Creating Splines	132
▪ Editing Splines	146

## Introduction

A spline generator (refer to the paper on Software Fragments, section on X-Splines, web page link <http://www.gk.dtu.dk/home/jab/software.html>) has been added to MEMS Pro to aid in fluidics device layout. This is an adaptation based on the lack of a spline primitive in MEMS L-Edit. Please read the web site to understand how the generator has been implemented. The beauty of the X-spline is that the user will have an intuitive way of dialing in spline behavior at each control point and it can be shown to go smoothly between the extreme of an interpolated curve to an approximated curve, passing through the "angled" curve (the original wire).

## Understanding Splines

A spline is defined by a set of vertices and shape factors. To each vertex is assigned a shape factor. The shape factors lie in the  $[-1, 1]$  interval.

- If the shape factor lies between  $[-1, 0[$  (where 0 does not belong to the set), the curve is interpolated in a geometrically continuous way.
- If the shape factor is 0, the vertices of the curve become angles. This shape factor allows one to recreate angled segments from curved elements.
- If the shape factor lies between  $]0, 1]$  (where 0 does not belong to the set), the created curve approximates the vertex.



The **Splines** option allows you to perform one of the following operations:

- Creating a spline
- Editing a spline

## Create Spline Dialog Box

The only dialog box used to create and edit splines is the **Create Spline** dialog box (Figure 52).

You access it by selecting either **Splines > Create** or **Splines > Edit** in the **MEMS Pro Toolbar**.

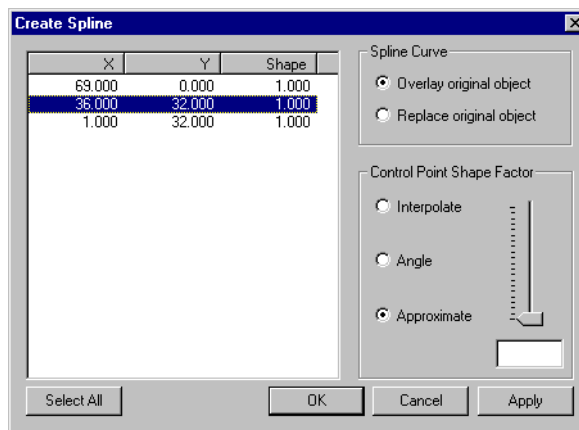


Figure 52: **Create Spline** dialog box

On the left side of the dialog box, the coordinates (in locator units) and the shape factor are displayed for each vertex (also called a control point) of the reference wire.

### Note

The vertices are displayed in a specific order depending on their numbers. You can view the numbers of the vertices by choosing **Tools > View Vertex Coordinates** in the **MEMS Pro Toolbar**. For more informations on this viewing feature, refer to Viewing Vertex Coordinates and Angles in Chapter 5 - MEMS Pro Utilities.

On the right side, the behavior of the curve at each control point can be manipulated either using the slide bar or the radio buttons. Whether using the slide bar or the radio buttons, one can change the behavior at a control point to be an approximate, angled or interpolated spline.

The parameters of the **Create Spline** dialog box are described in the following table:

Parameter	Default Value	Description
Interpolate	-1.00	Allows interpolation of a reference segment (the value ranges from -1 to 0)
Angle	0.00	Allows creation of angles on a curved object (the only available value is 0)

Parameter	Default Value	Description
Approximate	1.00	Allows approximation of the angle of a reference segment (the value ranges from 0 to 1)

You also have the possibility to decide whether you want to overlay the original object with the spline or replace the original object by the spline using the **Overlay original object** and **Replace original object** radio buttons of the **Spline Curve** area.

**Note**

---

The shape factor values range from [-1, +1].

---

# Creating Splines

## Creating Splines from Angled Wires

To create a spline from an angled wire, perform the following operations:

- ☑ Choose the **Selection** icon of the **Drawing** toolbar of MEMS Pro.
- ☑ Select the angled wire (often referred to as the "reference wire") (Figure 53) by clicking on it.

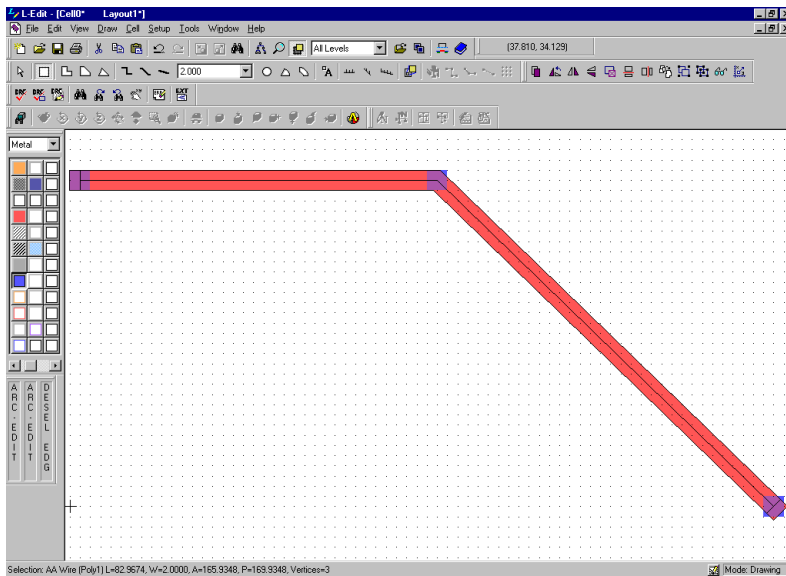


Figure 53: Selecting the reference wire

- ☒ Choose **Splines > Create** in the MEMS Pro Palette.

The **Create Spline** dialog box appears (Figure 54).

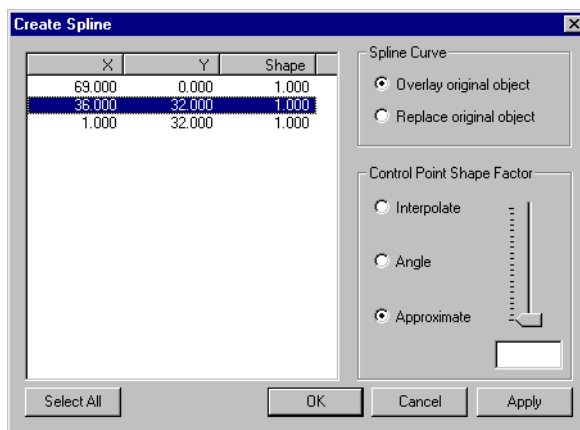


Figure 54: **Create Spline** dialog box

### *Interpolation*

A shape factor between  $[-1, 0[$  applied to the second vertex of the selected reference wire (Figure 53) will cause the shape to behave like an interpolation spline.

- ☒ Select the second vertex on the left side of the **Create Spline** dialog box.

- ☒ Select the **Interpolate** radio button in the **Control Point Shape Factor** box.
- ☒ Choose whether you want to replace the original object with the new one or not.
- ☒ Click **OK**.

The interpolated spline appears in the L-Edit window (Figure 55). It goes through all the control points, even the second one.



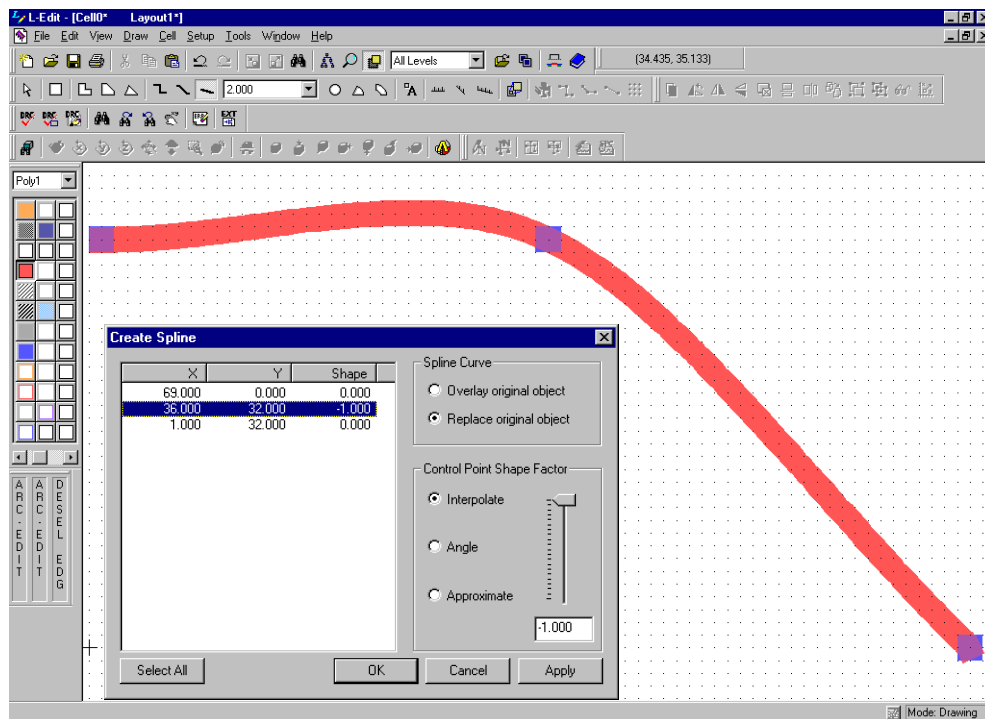


Figure 55: Interpolation spline

## *Approximation*

A shape factor between ]0,1] applied to the second vertex results in an approximation spline. The created curve approximates the second vertex of the reference wire.

- ☒ Select the second vertex on the left side of the **Create Spline** dialog box.
- ☒ Select the **Approximate** radio button in the **Control Point Shape Factor** box.
- ☒ Choose whether you want to replace the original object with the new one or not.
- ☒ Click **OK**.

The approximated spline appears in the L-Edit window (Figure 56). It does not go through the second vertex.

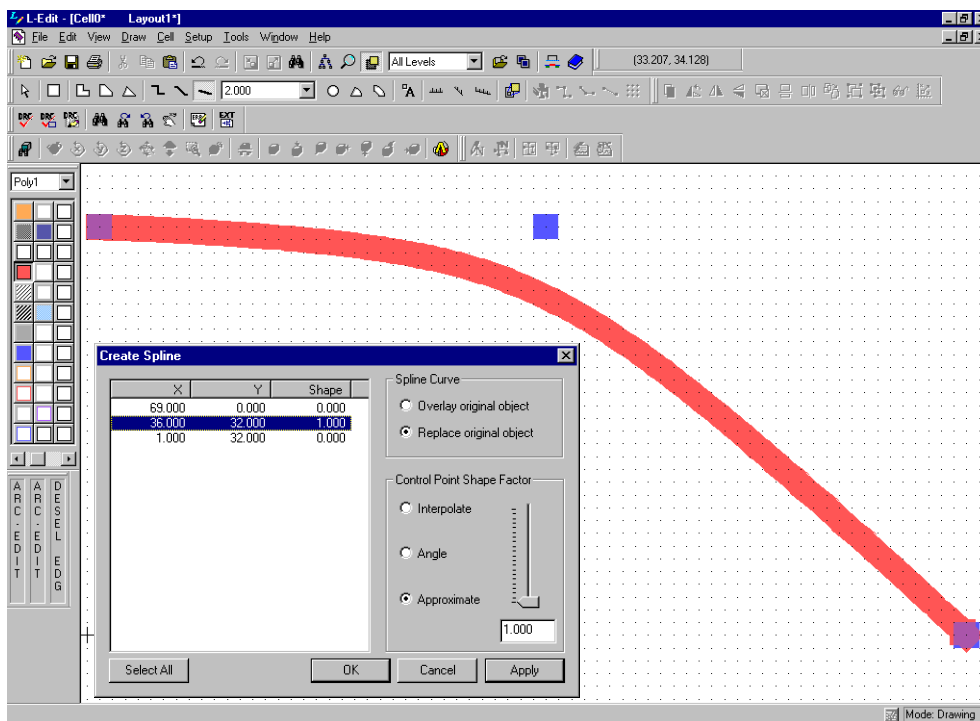


Figure 56: Creating an approximated spline

## Re-creating Angled Wires

A shape factor of 0 applied to the second vertex of a spline may be used to create (or re-create) an angled vertex. Thus, you can re-create the reference wire from the approximated or interpolated curve.

Perform the following operations:

- ☒ Select the spline.
- ☒ Choose **Splines > Create** in the **MEMS Pro Toolbar**.
- ☒ Select the **Angle** radio button of the **Control Point Shape Factor** box.
- ☒ Click **OK**.

The angled wire is created (or re-created) from the spline (Figure 57).

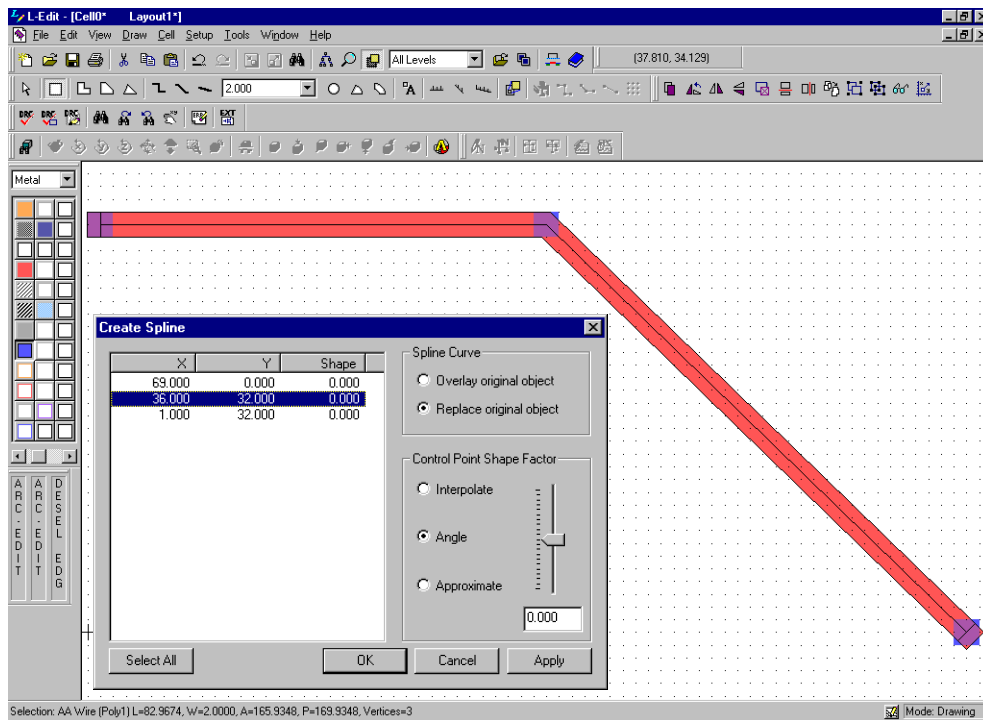


Figure 57: Angled wire

**Warning**

---

*The resultant curved wire is realized with a polygon since L-Edit does not implement a spline primitive. This has the downside of forcing you to start over if you click **OK** and then change your mind.*

---

## Creating Splines from Polygons

A more impressive possibility is that splines can be generated from polygons.

To view the advantages of this feature, perform the following steps:

- ☒ Create a five vertex polygon (Figure 58).

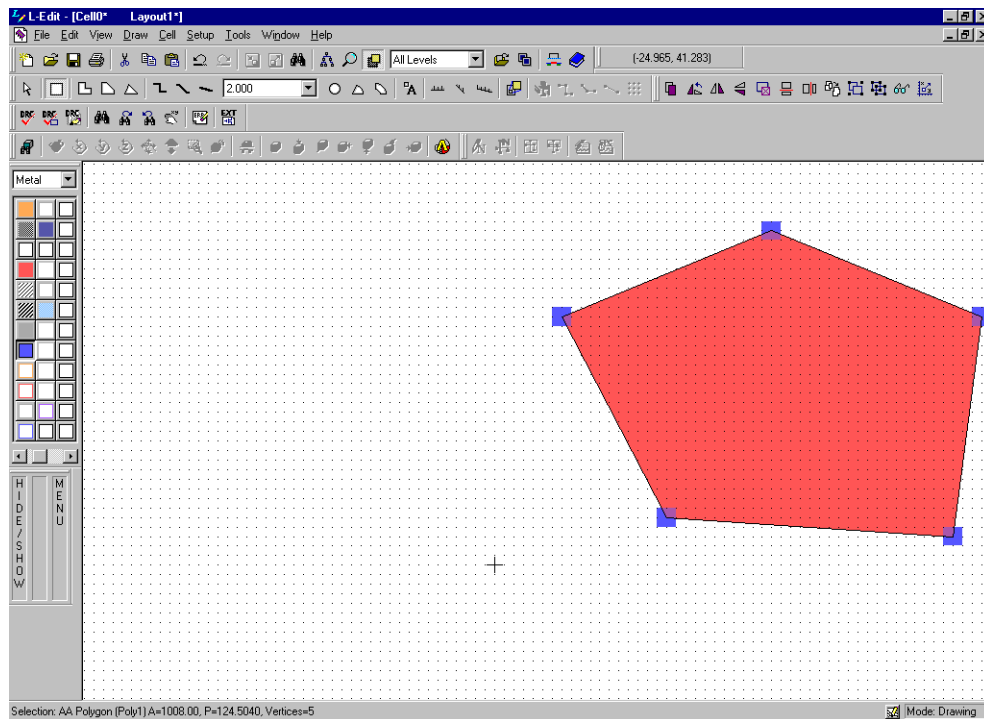


Figure 58: Polygon

- ☑ Select the polygon.
- ☑ Select **Splines > Create** in the **MEMS Pro Toolbar**.

The **Create Spline** dialog box appears.

- ☑ Modify the shape factors of all the vertices so that the curve approximates each vertex (Figure 59).

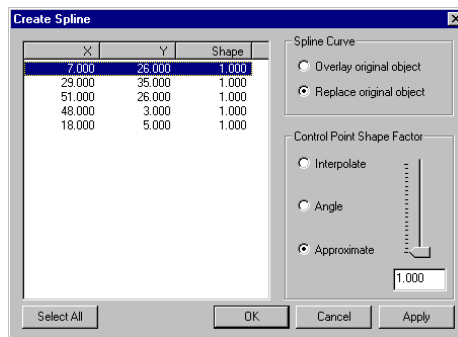


Figure 59: Modifying the vertex shape factors

- ☑ Click **OK**.



The created rounded shape of the polygon appears (Figure 60).

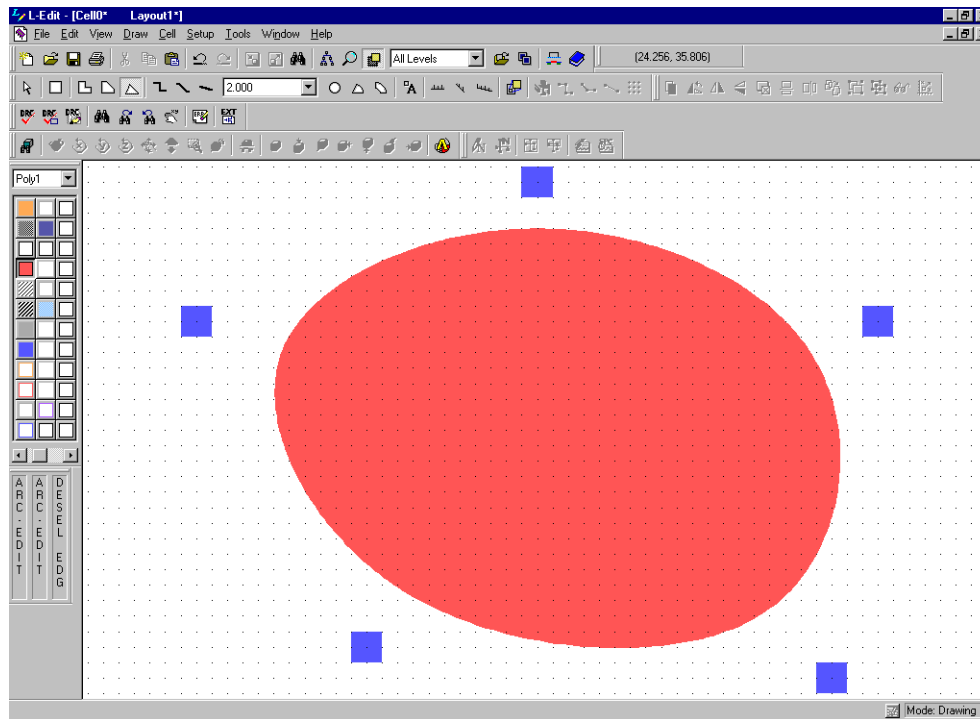


Figure 60: Rounded shape of the polygon

You can choose to either interpolate or approximate the vertices of the polygon. In the case of an interpolation, the curved resulting shape goes through all the vertices of the polygons. In the case of an approximation, the curved resulting shape does not go through any vertex of the original polygon.

## Editing Splines

A spline editor has been added to MEMS Pro to help in fluidics device layout.

To edit a spline, perform the following operations:

- ☒ Select the spline you want to modify.
- ☒ Choose **Splines > Edit** in the MEMS Pro Palette.

The **Create Spline** dialog box (Figure 52) appears. Follow the same method as the one used to create splines.

### Note

---

It is the same dialog as the one used for the creation of splines.

---

Using the dialog box radio buttons and/or slide bar, you can change the behavior of the existing spline to come back to the original shape (an angled wire) or to apply the opposite shape factor.

# 5 MEMS Pro Utilities

---

▪ Introduction	148
▪ Running Macros in L-Edit	149
▪ Generating Polar Arrays	150
▪ Generating Holes in a Plate	154
▪ Viewing Vertex Coordinates and Angles	157
▪ Approximating All-angle Objects	164
▪ Generating Concentric Circles	168

# Introduction

The MEMS Library contains a host of macros that facilitate the MEMS layout design process. In this chapter, we describe where these macros are located and how to use them.



## Running Macros in L-Edit

The macros described in this chapter should have been loaded automatically at start-up and bound to the **Tools >** menu in the **MEMS Pro Toolbar**. You may confirm whether the macros have been loaded by starting L-Edit, selecting the **Tools >** menu and looking for them.

### Loading the Macros

If the macros have not been loaded automatically, you may load them manually using the procedure below:

- ☑ Start L-Edit.
- ☑ Select **Tools > Macro** in the main L-Edit menu bar to access the **Macro** dialog.
- ☑ Click **Add** and load the macro file **<install directory>\memslibs\MEMSPHysical.dll**.
- ☑ To run a macro, either select the macro from the **Macros** list and click **Run**. Alternatively, you may access the macro from the **Tools** menu.

# Generating Polar Arrays

## Description

The **Polar Array** function allows you to generate multiple instances of a selected cell and to place them in an arc.

The **Polar Array** function depends on three parameters:

- The center of the polar array
- The number of desired copies
- The total angle for the copies

## Accessing the Function

- ☒ To access this function, select **Easy MEMS > Polar Array** in the **MEMS Pro Palette**.

The **Polar Array** dialog box opens.

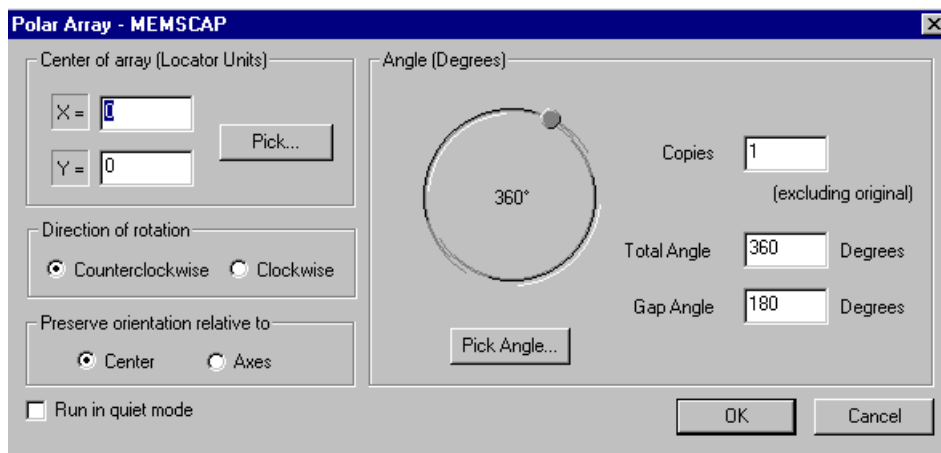


Figure 61: **Polar Array** dialog



## Parameters

The following table provides a description of the **Polar Array** dialog box options

Parameter	Default Value	Description
Copies	1	Number of copies in the polar array. This number does not include the original.
Center of array	X = 0, Y = 0	Center of the polar array (in locator units). You may enter the X and Y values in the edit fields or click the <b>Pick</b> button to select the center by clicking in the layout window. Picking a center displays a center mark in the layout.
Total Angle	360	Total angle according to which the polar array will be performed. You may choose between specifying the <b>Total Angle</b> or the <b>Gap Angle</b> .
Gap Angle	180	Angle separating the various occurrences of the device to copy. You may choose between specifying the <b>Total Angle</b> or the <b>Gap Angle</b> .

Parameter	Default Value	Description
Pick Angle		Clicking the <b>Pick Angle</b> button allows you to select the angle by selecting two points in the layout window. The angle is calculated between the 2 points and the center point. Picking an angle displays the center and angle text marks with a pie wedge sweeping out the angle.

## Generating Holes in a Plate

The **Plate Release** feature is a new MEMS Pro feature. To ensure the complete release of wide plates, holes must be cut out of the plate to allow the etching of the sacrificial oxide layer placed underneath the plate. Placing these holes can be a very time-consuming task that adds no significant value to the design; however, it is a necessary step to ensure manufacturability. The **Plate Release** utility dramatically shortens the process of adding these holes to plates by automatically generating them according to your options settings. You can use this tool to add dimples as well.

- ☒ To use the Plate Release function, select one shape (either on the poly1 or on the poly2 layer).

### Note

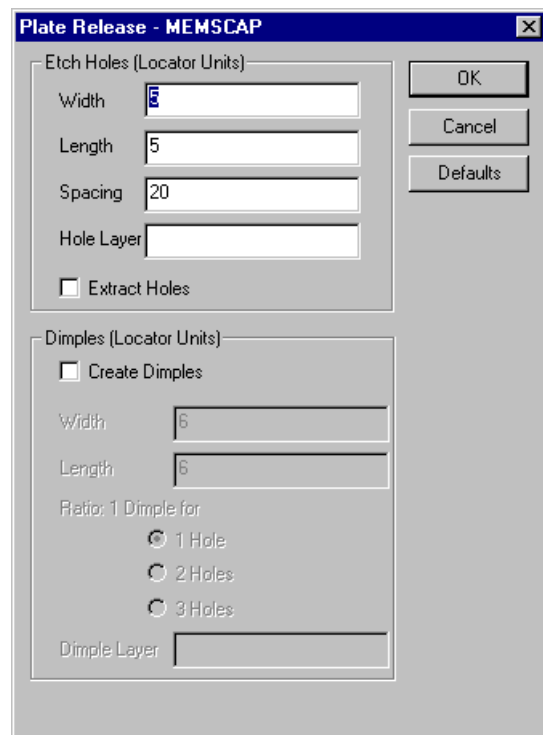
---

You have to select either a poly1 or a poly2 layer. If no shape is selected, an error message appears.

---

- ☒ Then, select **Easy MEMS > Plate Release** in the **MEMS Pro Palette**.

The Plate Release dialog box appears.

Figure 62: **Plate Release** dialog box

The following table explains the parameters used in the **Plate Release** dialog box and their explanation.

Parameter	Default Value	Description
Width	5	Width of the created holes
Length	5	Length of the created holes
Spacing	20	Space between the created holes
Extract Holes		Extract the holes from the shape
Create Dimples		The <b>Create Dimples</b> checkbox allows you to create dimples in addition to the holes

## Viewing Vertex Coordinates and Angles

Four new MEMS Pro features allow you to view the coordinates and angles of the vertices of selected objects:

- **View Vertex Coordinates**
- **View Vertex Angles**
- **View Vertex Information**
- **Clear Vertex Information**

### Note

These features are quite useful for the creation and edition of splines (refer to Chapter 4 - Splines).

## Viewing Vertex Coordinates

To view the vertex coordinates of a flat object (box, polygon, wire), perform the following operations:

- Select the object for which you want to view the vertex coordinates.
- Select **Tools > View Vertex Coordinates** in the MEMS Pro Palette.

The number and coordinates (in locator units) of each vertex are displayed on the layout as port text on the ruler layer at the corresponding vertices. The size of the port text is the default port text size of the Ruler layer. The text is displayed according to the following format:

vertex-number (X\_coordinate, Y\_coordinate)

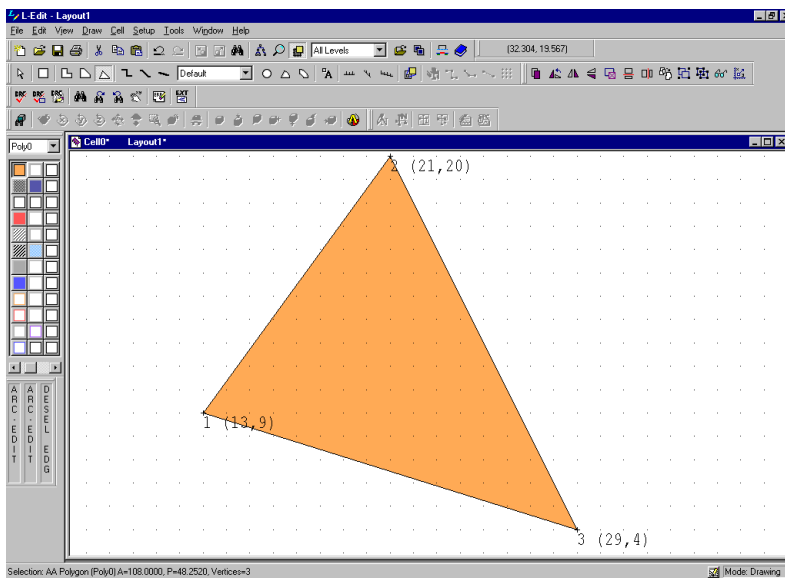


Figure 63: Viewing vertex coordinates

This information remains visible until the **Clear Vertex Information** option is issued.

## Viewing Vertex Angles

To view the vertex angles of a flat object (box, polygon, wire), perform the following operations:

- Select the object for which you want to view the vertex angles.
- Select **Tools > View Vertex Angles** in the **MEMS Pro Toolbar**.

The number and angle (in degrees) of each vertex are displayed on the layout as port text on the Ruler layer at the corresponding vertices. The size of the port text is the default port text size of the Ruler layer. The text is displayed according to the following format:

vertex\_number (vertex\_angle)



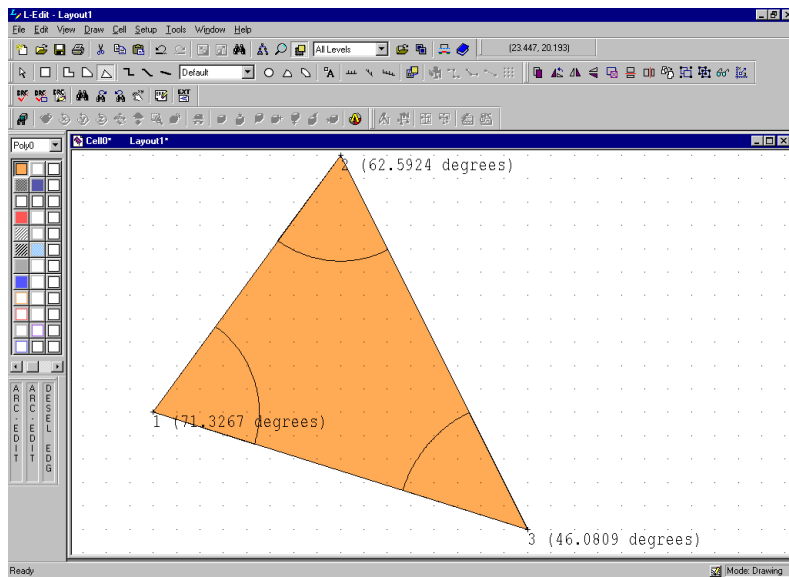


Figure 64: Viewing vertex angles

These information remain visible until the **Clear Vertex Information** option is issued.

## Viewing Vertex Information

To view the vertex information (number, coordinates, and angle) of a flat object (box, polygon, wire), perform the following operations:

- Select the object for which you want to view the vertices angles.
- Select **Tools > View Vertex Information** in the MEMS Pro Palette.

The number, coordinates (in locator units) and angle (in degrees) of each vertex are displayed on the layout as port text on the Ruler layer at the corresponding vertices. The size of the port text is the default port text size of the Ruler layer. The text is displayed according to the following format:

vertex\_number (X\_coordinate, Y\_coordinate) (vertex\_angle)

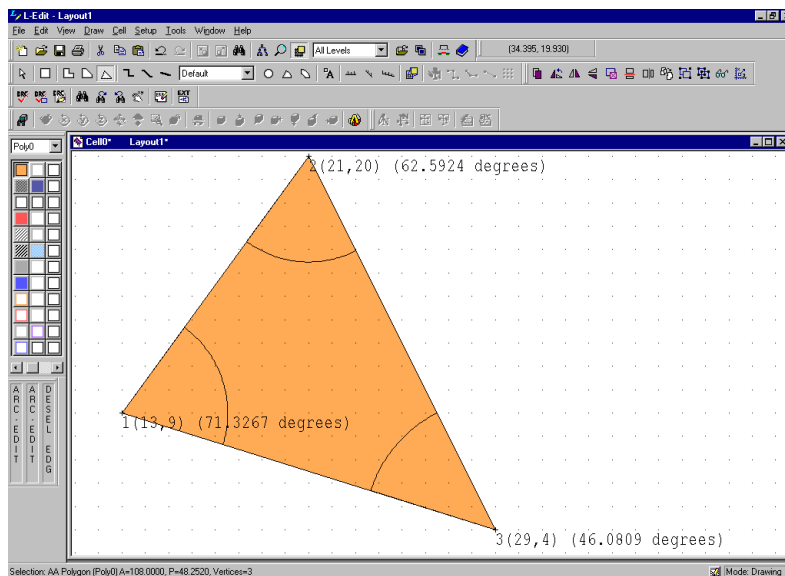


Figure 65: Viewing vertex information

These information remain visible until the **Clear Vertex Information** option is issued.

## Clearing Vertex Information

To remove the vertex information from the layout view of an object, perform the following operations:

- Select the object for which you want to clear the vertex information
- Select **Tools > Clear Vertex Information** on the MEMS Pro Palette.

A dialog box prompting you to confirm the removal of the vertex information of the selected object appears. Confirm your intention of removing the vertex information of the selected object by clicking **OK**. The vertex information of the selected object disappears.

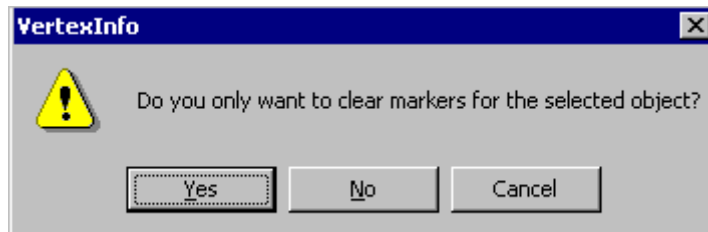


Figure 66: **VertexInfo** Confirmation dialog

---

### Note

If you click **No**, the vertex information of all the objects is cleared.

---

# Approximating All-angle Objects

## Description

**Approx.dll** generates 90° and 45° approximated polygons for all-angle objects within a cell. The macro operates on the entire hierarchy of the design. The user may choose to perform approximation, approximation and cross-section view, approximation and design rule check, or approximation and extract.

## Accessing the Macro

To access this macro, select any of the following commands: **Tools > MEMS Approx**, **Tools > MEMS CSV**, **Tools > MEMS DRC**, or **Tools > MEMS Extract**.

Parameter	Default Value	Description
<b>MEMS Approx</b>		Available approximation types are <b>Manhattanize</b> and <b>Bostonize</b> . <b>Manhattanize</b> generates 90° polygon approximations. <b>Bostonize</b> generates 90° and 45° polygon approximations.
<b>MEMS CSV</b>		Generates a cross-section view after a <b>Bostonize</b> approximation.

Parameter	Default Value	Description
<b>MEMS DRC</b>		Initiates a design rule check after a <b>Bostonize</b> approximation.
<b>MEMS Extract</b>		Extracts the layout after a <b>Bostonize</b> approximation.

These commands must already be loaded, as described previously. Once any of them are invoked, the **All Angle Approximation** dialog will appear.

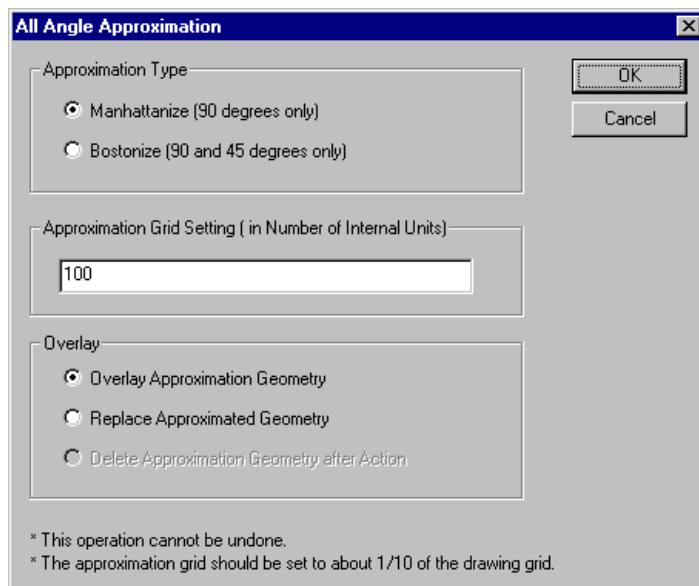


Figure 67: **All Angle Approximation** dialog

## Parameters

Parameters	Default Value	Description
<b>Approximation Type</b>		Choose between <b>Manhattanize</b> (90° only) and <b>Bostonize</b> (90° and 45°). This parameter only applies to the <i>MEMS Approx</i> action.
<b>Approximation Grid</b>		Grid for approximation in number of internal units. The default grid is 100 since most technology files use 1000 internal units per locator unit. Choosing 1/10 of the technology unit for this setting is sufficient for most layouts. Finer grid selections will adversely affect execution time.
<b>Overlay</b>		There are three overlay options: <b>Overlay Approximation Geometry</b> , <b>Replace Approximated Geometry</b> , or <b>Delete Approximation Geometry after Action</b> . The first option overlays the approximated objects on top of the existing all-angle objects. The second option replaces the existing all-angle objects with the approximated objects. The third option is enabled only for <b>DRC</b> and <b>Extract</b> ; it deletes approximated objects when the action is complete.



# Generating Concentric Circles

## Location

The **ccircle.dll** library is located under the following directory:

**<install directory>\memslibs\ccircle.dll**

## Description

**ccircle.dll** generates concentric circles on the current layer in L-Edit. Dimensions and the fill-type of the circles must be submitted in an ASCII file.

## Accessing the Macro

To access this macro, select **Tools > Concentric Circles**.

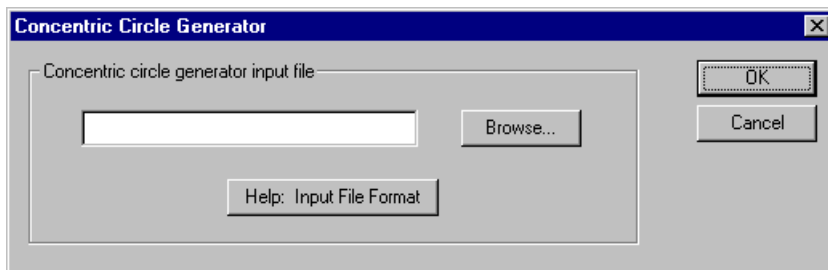


Figure 68: **Concentric Circle Generator** dialog

## Parameters

Parameters	Default Value	Description
Input File		Input file name.

## Input File Format

**Ccircle.inp** is a sample input file.

## *Syntax*

**.group <groupname> CCIRCS**

**<circlename> <radius> <filltype>**

**.**

**.**

**.**

**.endg <groupname>**

Comment lines begin with the \* character.

## *Example*

**.group AGroup CCIRCS**

**circ 10000 1**

**circ2 20000 0**

**circ3 35000 1**

**.endg AGroup**

## Parameters

Parameters	Default Value	Description
<b>groupname</b>		Any sequence of characters is allowed for the <b>groupname</b> except \ and white space.
<b>radius</b>		The <b>radius</b> of circle in Internal Units. The radii of circles found in a group block must be of increasing size. See the example above.
<b>filltype</b>		The value of the first circle's <b>fill type</b> indicates whether the circle is to be filled or left blank. If fill type of the first circle is the value 1, then the circle is filled. That is, the space between the center and the circumference of the circle will be assigned a layer; outside the circle's circumference, the region will be left blank. Circles of greater radius will be drawn, as defined in the following geometry definition statements. The fill types of these circles will be ignored; alternating between filling and not filling the areas between the subsequent circles in the group. If the first circle has a fill type of value zero, 0, then the area between the center and the circumference of the circle will be left blank. The area between subsequent circles will be alternately filled and left blank.

## 6 3D Modeler

---

▪ Introduction	173
▪ Accessing 3D Models	182
▪ Defining Colors for 3D Models	186
▪ Viewing 3D Models from Layout	188
▪ 3D Model View User Interface	190
▪ Viewing a Cross-section	215
▪ Deleting 3D Models	218
▪ Exporting 3D Models	220
▪ Linking to ANSYS	223
▪ Editing the Process Definition	225
▪ 3D Modeler Error Checks	252

## Introduction

The 3D Modeler emulates the geometric effects of the fabrication process on a wafer from its mask layout and process definition. Once created, the 3D model is displayed in an L-Edit window. Additional views of the model in separate windows may also be created. These views can then be rotated, translated and scaled.

The 3D Modeler may be used to catch errors in the layout and fabrication process before submitting a design to a foundry. It is an essential tool for generating input files for 3D device analysis. 3D models created in the MEMS Pro 3D Modeler may be exported for direct use with third party finite element and boundary element analysis tools, including those available from ANSYS, CFDRC, Coyote Systems and Hewlett Packard.

Four examples of 3D models of MEMS devices (a thermal actuator, a rotary motor, an accelerometer, and a diaphragm) are presented on the following pages.

### MCNC MUMPs Thermal Actuator

The model below is of a thermal actuator designed for the MCNC MUMPs 3 layer polysilicon surface micromachining process. The thin arm has greater resistance than the thick arm. Therefore, when a voltage is applied to the contact pads, the thin arm heats more rapidly and to a higher temperature than the thick

arm. The larger thermal expansion of the thin arm causes the tip of the actuator to deflect upward.

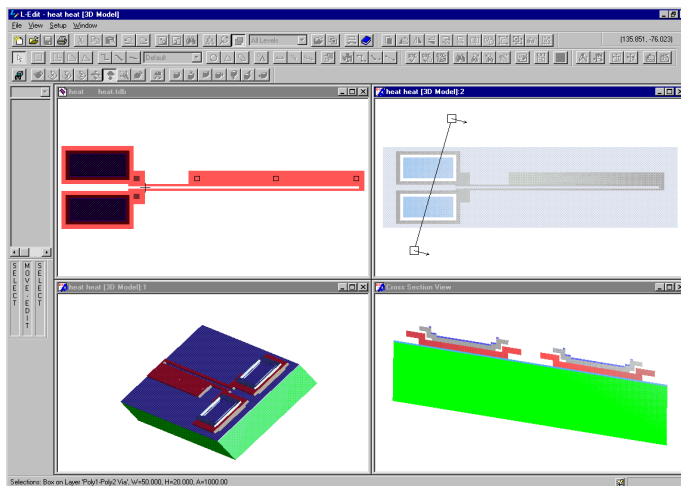


Figure 69: Various views of the thermal actuator

In Figure 1, four views of the thermal actuator are displayed (clockwise from the upper left): a layout view, a top view of the 3D model with a cross-section line, a view of the cross-section, and a rotated view of the 3D model.

The .tdb file containing the example of this thermal actuator is <install directory>\Examples\3DModel\mumps\heat\heat.tdb.

## MCNC MUMPs Rotary Motor

The device modeled below is a rotary side drive motor designed for the MCNC MUMPs 3 layer polysilicon surface micromachining process. The motor has twelve stators and eight rotors. Tangential electrostatic forces tending to align the rotor poles with the excited stator poles cause the hub to rotate.



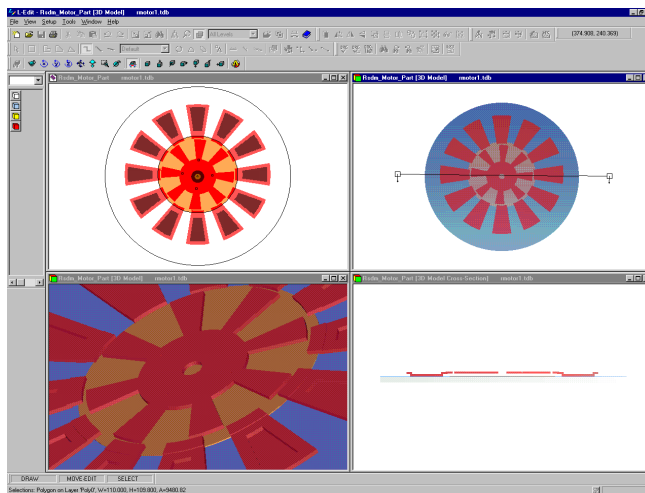


Figure 70: Various views of the rotary motor

In Figure 2, four views of the rotary side drive motor are displayed (clockwise from the upper left): a layout view, a top view of the 3D model with a cross-section line, a view of the cross-section, and a rotated close-up view of the 3D model.

The .tdb file containing the rotary side drive motor example is <install directory>\Examples\3DModel\mumps\RotMotor\motor.tdb.

## Analog Devices iMEMS ADXL Accelerometer

Shown below is an accelerometer designed using Analog Devices' iMEMS process. The iMEMS process is a surface-micromachining process that enables the fabrication of a polysilicon MEMS device and BiCMOS interface circuitry on a single chip. The accelerometer is a center plate that is suspended between a pair of springs. Comb fingers attached to the two sides of the plate create a differential capacitor with the set of fixed outer fingers. The interface circuitry (not shown) creates a feedback control system that applies sufficient electrostatic force to balance the effects of acceleration. The feedback is used to measure the acceleration.

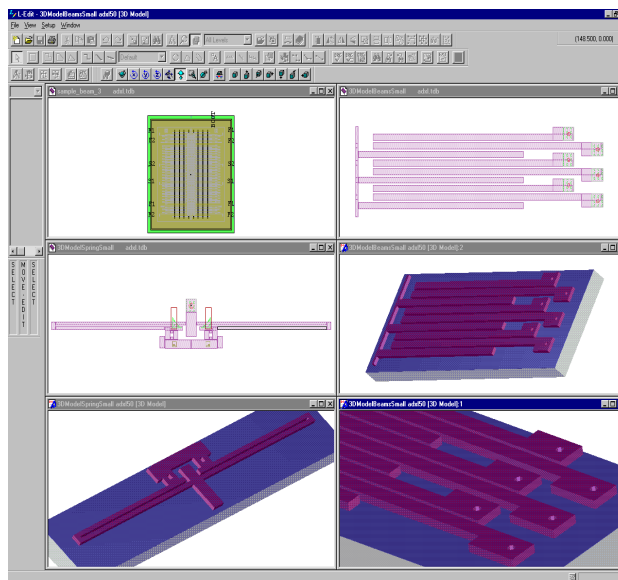


Figure 71: Various views of the ADXL accelerometer

In Figure 3, six views of the accelerometer are displayed (clockwise from the upper left): a layout view of the entire sensor, a layout view of a set of comb

fingers, two views of the 3D model of the comb fingers, a 3D model of one of the springs, and a layout view of a spring.

The .tdb file containing the example of this accelerometer is <install directory>\Examples\3DModel\adimems\ adimems.tdb.

## Bulk Micromachined Diaphragm

Shown below is a diaphragm suspended by four beams over a pit created by a backside etch of a wafer. Such devices may be designed to sense pressure by placing piezoresistors at the center of the diaphragm edges.

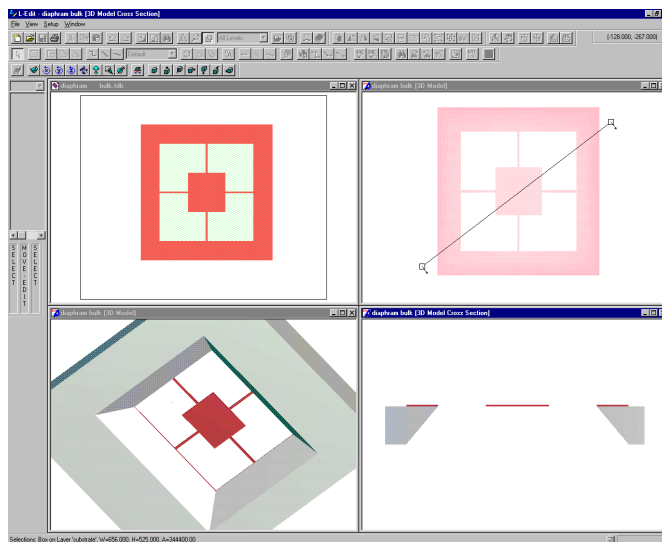


Figure 72: Various views of the bulk micromachined diagram

In Figure 4, four views of the diaphragm are displayed (clockwise from the upper left): a layout view, a top view with cross-section line, a cross-section view, and a rotated view of the 3D model from beneath.

The .tdb file containing the example of the bulk micromachined diagram is <install directory>\Examples\3DModel\bulk\bulk.tdb.

# Accessing 3D Models

## 3D Model Input

3D models of MEMS devices can be created, viewed, and manipulated within L-Edit. Both L-Edit mask layout and a process definition are required to accomplish this task. Process definitions summarize the geometric effect of the fabrication steps used to construct a device. These definitions are parameterized in geometric terms (such as etch depths and etch angles), and not in processing terms (such as time of immersion or ambient temperature).

The 3D Modeler can view models stored in **SAT** (.sat) format but it cannot edit them.

Process definitions can be read from a text (.pdt) file or they can be entered manually through the **Edit Process Definition** dialog. For more information about *defining processes*, see Process Definition on page 352.

## 3D Modeler Output

The 3D model may be stored with mask layout in an L-Edit .tdb file. The 3D model may also be exported as a **SAT** (sat) or **ANF** (.anf) file.

The **.sat** file format is commonly used to exchange data between 3D model visualization and analysis tools. **SAT** is a standard industry format, and is accepted by many tools including AutoCAD, ANSYS, Ansoft HFSS, Maxwell 3D, ABAQUS, and MSC/NASTRAN and those from CFDRC and Coyote Systems.

The **.anf** file format is the **ANSYS Neutral Format**. **ANF** files can be directly imported into ANSYS. The details of converting **SAT** files to the **ANF** file format depend on your operating system.

Under Windows 95, MEMS Pro users must export their solid models in **ANF** format if they wish to use ANSYS. MEMS Pro uses the ANSYS connection module called *The ANSYS Connection Product for SAT* to write the **ANF** file that describes your model. You must have this connection module installed in your ANSYS directory to accomplish this task.

Under Windows NT and UNIX, MEMS Pro users who wish to use ANSYS have the option of exporting their files in either **ANF** or **SAT** format. If you choose to export to **ANF** format, MEMS Pro will invoke *The ANSYS Connection Product for SAT* as you export the file. If you choose to export your model from MEMS Pro in **SAT** format, ANSYS will invoke *The ANSYS Connection Product for SAT* as it reads the **SAT** file. In either case, you must have *The ANSYS Connection Product for SAT* installed in your ANSYS directory.



## Accessing the 3D Tools

3D model tools may be accessed from the L-Edit layout menu.

The **Edit Process Definition**, **View 3D Model**, **Delete 3D Model**, and **Export 3D Model** options may be accessed through the **3D Tools** button of the **MEMS Pro Toolbar**.

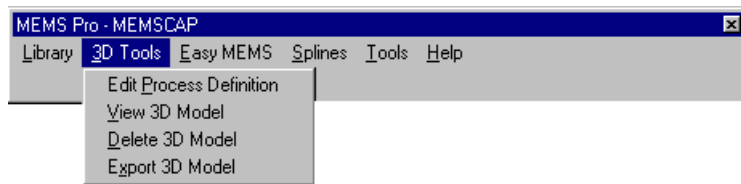


Figure 73: Accessing the **3D Tools** option

The **View 3D Model**, **Delete 3D Model**, and **Export 3D Model** options may also be accessed from the **Design Navigator**'s context-sensitive menu, which is

reached by a right-click while in the **Design Navigator** window. The **Design Navigator**, can be reached from the L-Edit **View** submenu.

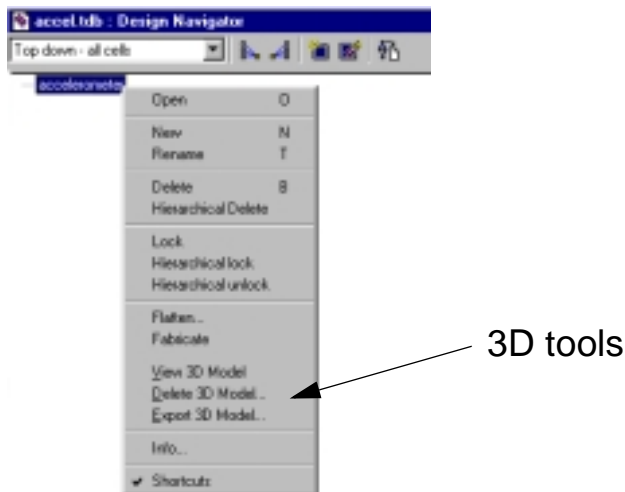


Figure 74: Accessing the **3D Tools** options using the context-sensitive menu

All commands available from the context-sensitive menu operate on the selected cell. Since the process definition is a file-wide property, it is not possible to access the **Edit Process Definition** dialog from this menu.

## Defining Colors for 3D Models

The setups for the standard fabrication processes have preset colors for the solid bodies that result from fabrication process steps; these colors are related to the mask layout color for the 3D models. It is not necessary to specially assign colors for the 3D models, but if you wish to define colors you may do so.

The color corresponding to each of the layer materials is determined by parameters set in L-Edit **Setup Layers** dialog.

The layout window must be active for you to access the color setup. If the layout is active, the standard L-Edit layout menu bar will appear at the top of the window. If this is not the case, move the cursor to a layout window, and left-click to activate it.

From the L-Edit layout menu, **Setup > Layers** will invoke the **Setup Layers** dialog box. Click the **Rendering** tab. Select **3D Model** in the **Pass List** to show the current color settings for models generated from the selected layer.

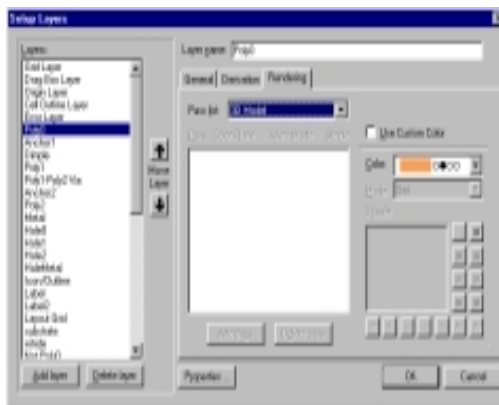


Figure 75: **Setup Layers** dialog

By default, the **Use Custom Color** check box is unchecked. To customize the 3D model colors, check **Use Custom Color**. The interface highlights the color of the first pass on the object **Pass List**, but you may choose alternate colors from the **Color** sample bar. You can select colors for each layer in your 3D model. Solid colors are available in MEMS Pro Version 3. Stipple patterns are not.

## Viewing 3D Models from Layout

Model viewing is launched by selecting **3D Tools > View 3D Model** in the **MEMS Pro Toolbar**. If the model is up-to-date, you may immediately view it. If the model is new, or needs to be updated, the model is generated when the **View 3D Model** command is selected. You will be warned if the model is out-of-date. For more information on warnings and error messages, see 3D Modeler Error Checks on page 252.

During model generation, a progress bar will display the **Label** and **Process Step** number associated to each command as it is processed by the 3D Modeler. For more information on **Labels** and **Process Step** numbers, see Editing the Process Definition on page 225.

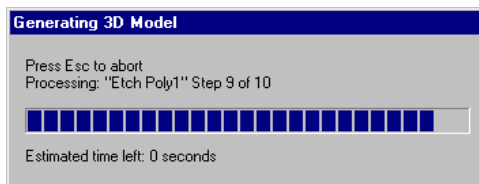



Figure 76: **Generating 3D Model** progress bar

An estimate of the time remaining to complete each step is reported below the progress bar. To abort 3D model generation, press the **Esc** key.

The model is displayed in an L-Edit **3D Model View** window with the  icon in the left corner followed by **Cellname [3D Model] Filename**. The initial view of the 3D model will be **Isometric** (that is, with equal X, Y, and Z-scales and the X, Y, and Z-axes drawn 120 degrees apart).

**Note**

---

For more information on *viewing the 3D models*, see 3D Model View User Interface on page 190.

---

# 3D Model View User Interface

## Application Elements

The graphical user interface for the **3D Model View** has six important screen components: the **Title Bar**, the **Menu Bar**, the **3D Model Tool Bar**, the **Palette**, the **Status Bar**, and the **Work Windows**.

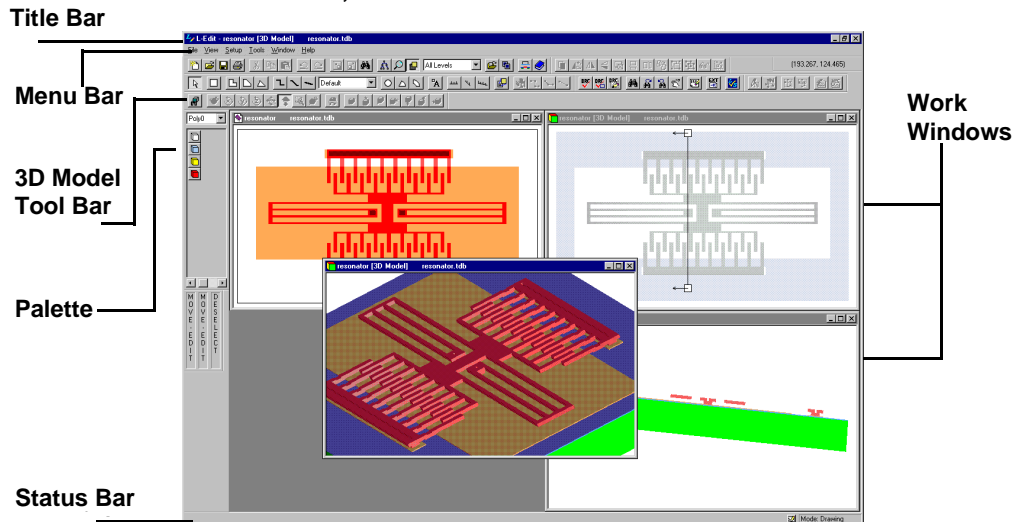


Figure 77: Graphical User Interface for the 3D Model View

The L-Edit **Locator** and **Mouse Button Bar** are inactive while in a **3D Model View**.

While in **3D Model View** mode, the left, center, and right mouse buttons offer shortcuts to three view commands:

- **Ctrl+Left** activates the **Orbit View**
- **Ctrl+Center** activates the **Pan View**
- **Ctrl+Right** activates the **Drag-Zoom View**.

### Note

---

For two-button mice, the **Pan View** may be accessed by clicking the **Ctrl+Alt+Left** combination.

---

See View Menu on page 195 for more information.

## Title Bar

When a **3D Model View** is active, the L-Edit application title bar indicates the current cell name, active window type in square brackets, and the file name: ***Cellname [3D Model] Filename***. Further, the application window can be reduced to an icon, zoomed, resized, moved, or closed from this title bar.



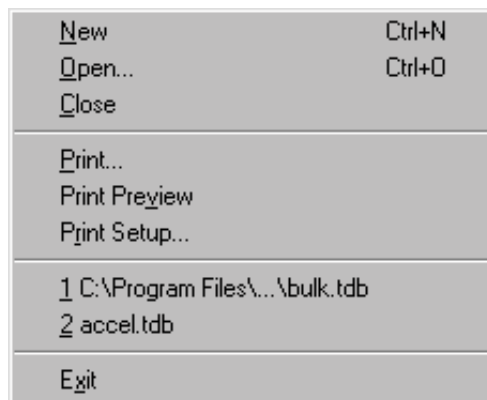
## Menu Bar

The **Menu Bar** refers to six **3D Model View** menus. The menus can be opened to show available commands by clicking the menu bar or by pressing the keyboard shortcuts indicated below:

<b>File</b>	<b>Alt+F</b>	Commands for creating, opening, saving, and printing files.
<b>View</b>	<b>Alt+V</b>	Commands for expanding, contracting, and shifting the view.
<b>Tools</b>	<b>Alt+T</b>	Commands for viewing, deleting, and exporting 3D models and editing process definitions.
<b>Setup</b>	<b>Alt+S</b>	Commands for customizing interface elements and program functions.
<b>Window</b>	<b>Alt+W</b>	Commands for manipulating windows.
<b>Help</b>	<b>Alt+H</b>	Commands for invoking Tanner EDA help documentation.

### *File Menu*

The **File** menu contains commands for opening, saving, and printing files.

Figure 78: **File** menu options**New**

Invokes a dialog to create a new text or layout file.

**Open**

Opens an existing Tanner Database (**.tdb**) file, text files, or a solid model in **SAT** format. If a 3D model has been saved to this **TDB** file, the model can be brought up in a window with the **View 3D Model** command.

If you wish to open a **SAT** file, select **File > Open**, and choose **3D Model Files (.sat)** in the **Files of type** field. Recall that the view of a **SAT** file can be manipulated by the 3D Modeler, but the model itself cannot be edited.

**Close**

If the **3D Model View** contains an internally generated 3D model, then **Close** will close all the windows for the model and leave the **.tdb** file, from which it was generated, open.

If a model from an external (**.sat**) file is under examination, **Close** will simply close that file and close the window.

**Print, Print Preview, Print Setup**

These commands allow you to print and preview the contents of the **3D Model View** window, and to change printer and print settings.

**Recently Opened Files**

The most recently opened files are listed here. If any **.sat** files have been accessed, they will appear on the list as well as the Tanner EDA database and text files.

**Exit**

**Exit** will prompt to save changes and then quit L-Edit.

## View Menu

The **View** menu contains seven **Preset Views** and five interactive viewing options, namely, **Spin**, **Orbit**, **Rotate**, **Pan**, and **Zoom**.

There are also options for determining the look and content of the **Toolbars** and **Status Bars**.



Figure 79: **View** menu options

**Preset Views >**  
**Isometric,**  
**Top, Front,**  
**Right, Bottom,**  
**Back, Left**

As shown below, seven common viewing angles are available with the **Preset View** menu item. As you move from one command to another, only the viewing angle changes, not the magnification.

**Preset Views >** As shown below, seven common viewing angles are available with the **Preset View** menu item. As you move from one command to another, only the viewing angle changes, not the magnification.

**Isometric,  
Top, Front,  
Right, Bottom,  
Back, Left**

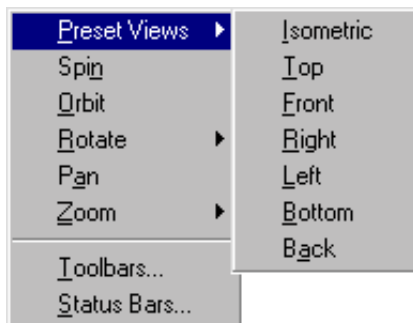


Figure 80: Options of the **Preset Views** menu

The **Isometric** view has equal X, Y, and Z-scales and has the X, Y, and Z-axes drawn 120 degrees apart.

For the **View** tools, the center of the 3D model is the origin of the X, Y, and Z-axes, the **Top** view is from above the object, parallel to the X-Y plane. The **Front** view is from the positive X direction looking back at the object and parallel to the Y-Z plane. The **Right** view is from the positive Y direction looking back at the object parallel to the X-Z plane. Likewise, the **Left** view is from the negative Y direction, the **Bottom** view is from the negative Z direction, and the **Back** view is from the negative X direction.




### Spin

This selection will cause the 3D model to rotate around the Z-axis for one complete revolution. Note that if your 3D object is symmetrical about the Z-axis, it will appear to revolve twice.

### Orbit

**Orbit** gives you an arbitrary view of the 3D model. This command causes the model to rotate around its center (thus accomplishing angular motion along all three axes simultaneously) as you drag the mouse, and can be accessed through a **Ctrl+Left** click keyboard-mouse combination.



## Rotate

**Rotate** refers to a motion about a single axis. As shown below, you may choose to rotate the view about the **X-axis**, **Y-axis**, or **Z-axis**. In the diagram below, the **X-axis** is selected. Once **Rotate** is selected, the left mouse button must be clicked and the mouse must be moved to activate the command. Note that since the point of view, not the model, is shifted, moving the mouse to the left will appear to rotate the model to the right.

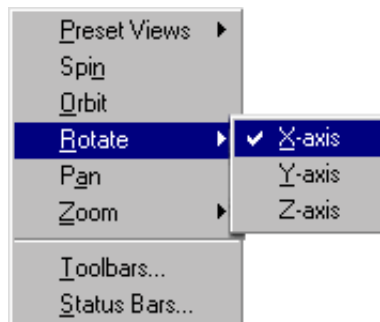


Figure 81: Selecting the X-axis as rotation axis

## **Pan**

**Pan** translates your view of the object while maintaining its orientation and magnification. If you move the mouse to the left, the window will follow and your view will move to the left. If you move the mouse to the top of the screen, your view will follow to the top of the screen.

**Pan** can be accessed through a **Ctrl+Center** click keyboard-mouse combination. For two-button mice, the combination **Ctrl+Alt+Left** click will access **Pan**.

## **Zoom**

As shown below, the **Zoom** command varies the magnification of the 3D model.



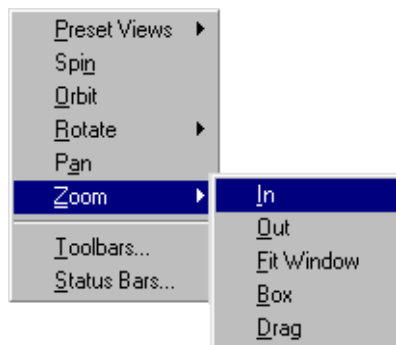


Figure 82: Selecting the zoom-in command

<b>Zoom &gt; In</b>	Magnification is increased by a factor of two.
<b>Zoom &gt; Out</b>	Magnification is decreased by a factor of two.
<b>Zoom &gt; Fit Window</b>	Magnifies the 3D model to fit the window.

- Zoom > Box** Left-click once to set one corner of the box, left click again to select the opposite corner of the box. The area bounded by the box will fill the entire **3D Model View** window.
- Zoom > Drag** Once you are in drag mode, you can smoothly zoom towards and away from the model by holding down the left mouse button and dragging the mouse forwards and backwards. Zooming towards the model increases the magnification; zooming away decreases the magnification.
- Toolbars** As shown below, the **3D Model View** toolbar may be shown or hidden. This same dialog box may be reached with L-Edit **View > Tools > Toolbars** command. Any of the toolbars shown below may be hidden by unchecking the box next to its name. Click **Close** to exit the dialog.

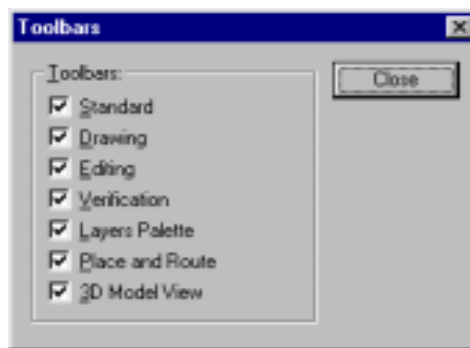


Figure 83: **Toolbars** dialog

### Status Bars

As shown below, you may show or hide the **Status Bar** while viewing 3D models. Note that the **Mouse Button Bar** and **Locator Bar** are not active while in **3D Model View**. You may remove a bar by unchecking the box to its left. Click **Close** to exit the **Status Bars** dialog.



Figure 84: **Status Bars** dialog

For more information on the **Status Bar**, see Status Bar on page 213.

## Tools Menu

The **Edit Process Definition**, **Regenerate 3D Model**, **Delete 3D Model**, and **Export 3D Model** commands can all be accessed from the **3D Tools** button of the MEMS Pro Palette.

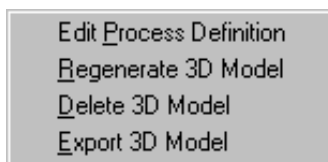


Figure 85: Options of the **3D Tools** menu

### **Edit Process Definition**

You may import and edit process definitions through this command. This menu command performs the same action as **Tools > 3D Tools > Edit Process Definition** described in Editing the Process Definition on page 225.

### **Regenerate 3D Model**

This menu command overwrites the current 3D model with a newly generated one. It performs the same action as **Tools > 3D Tools > View 3D Model** described in Viewing 3D Models from Layout on page 188, assuming that the model is not up-to-date and you chose to regenerate it.

**Delete 3D Model**

This command deletes the currently viewed 3D model and removes all of its open views.

**Export 3D Model**

This menu command allows you to export files in **SAT** or **ANF** format. It performs the same action as **Tools > 3D Tools > Export 3D Model** described in Exporting 3D Models on page 220.

*Setup Menu*

The **Setup Application** dialog can be reached through the **Setup** menu item on the **3D Model View** menu bar.

## Window Menu

The **Window** menu contains commands that are used to create and arrange multiple windows. These commands are **Cascade**, **Tile**, **Arrange Icons**, **Split Horizontal**, and **Split Vertical**.

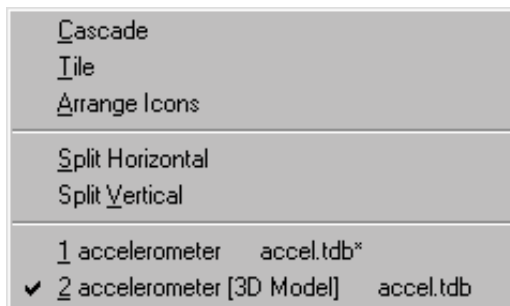


Figure 86: Options of the **Windows** menu

### Cascade

Arranges windows in an overlapping fashion, starting from the top left corner of the work area, so that the title bars are visible. The active window remains active (in front).

### Tile

Resizes all the open windows so that they do not overlap. Windows will appear in a row and column matrix.

<b>Arrange Icons</b>	Arranges any minimized window icons presenting rows starting at the bottom left of the work area.
<b>Split Horizontal</b>	Splits the active window horizontally and copies the 3D model view onto both panels. The 3D model or cross-section views may be independently manipulated.
<b>Split Vertical</b>	Splits the current window vertically and copies the 3D model view onto both panels. The 3D model view or cross-section views may be manipulated independently.
<b>Currently open files</b>	The last items on the <b>Windows</b> menu vary. Names of all the currently opened windows appear below the <b>Split Vertical</b> choice.

## *Help Menu*

Online versions of the standard L-Edit manuals as well as the *MEMS Pro User Manual* can be directly accessed from the **Help** menu. Information about the installation of L-Edit on your machine, including installed modules, version



number, memory allocation and how to contact technical support can be found by selecting **About L-Edit**.

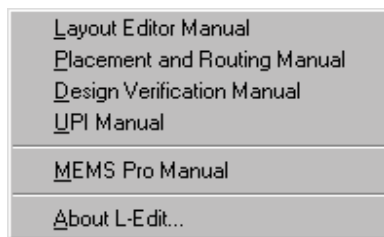


Figure 87: Options of the **Help** menu

## 3D Model Tool Bar

The **3D Model Toolbar** buttons represent the most commonly used viewing commands. See View Menu on page 195, Viewing a Cross-section on page 215, and Linking to ANSYS on page 223 for specific descriptions of each command listed below.



Figure 88: **3D Model View** Toolbar



**MEMS Pro Palette > 3D Tools > View  
3D Model**



**View > Orbit**



**View > Rotate > X-axis**



**View > Rotate > Y-axis**



**3D Model Cross-section View**



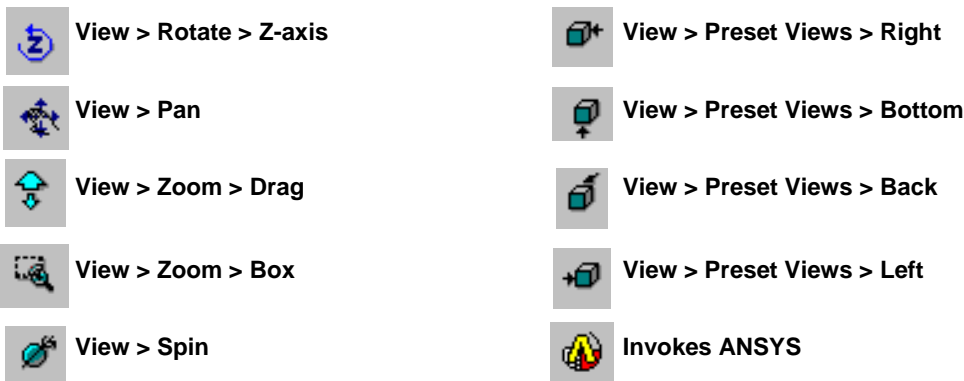
**View > Preset Views > Isometric**



**View > Preset Views > Top**



**View > Preset Views > Front**



The bottom icon on the right is not a **3D Model View** command. It is a hot link to ANSYS, a program that performs finite element and boundary element analyses. See the ANSYS Tutorial on page 176 for more information on ANSYS.

Note that in L-Edit layout mode the **3D Model View** and **ANSYS** buttons are both active, while in **3D Model View** mode, only the **ANSYS** button.

## Palette

The **3D Model View** palette is similar in look and function to the L-Edit layer palette, except that individual choices are displayed as cubes, not squares. The **3D Model View** palette displays the colors of the 3D bodies; these settings are

related to the mask layer of the same name. The palette contains only the 3D bodies present in the active 3D model.



Figure 89: **3D Model View** Palette

Hiding layers is particularly useful for obtaining a view of the interior of a 3D model. Layers may be hidden or shown using the **3D Model View** palette. To toggle between **hide** and **show**, center-click on the desired layer (for two-button mice, **Alt+Left** click).

**Hide** or **Show** layers is also available from a context-sensitive menu. Right-click on the icon corresponding to the layer you want to hide or show.

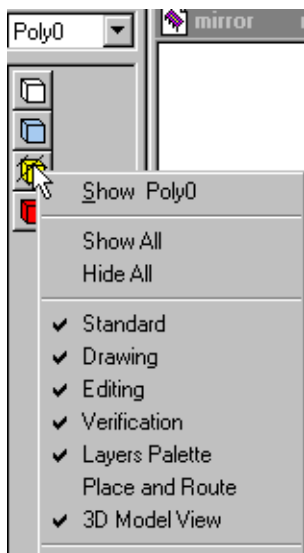


Figure 90: Context-sensitive menu

If the body is currently hidden, as in the diagram above, the icon corresponding to that layer will appear with hash marks. The context-sensitive menu will offer the option to **Show** that layer, followed by **Show All** and **Hide All** layers.

If the body is currently displayed, the icon will appear as solid color, and the context-sensitive menu will display the option to hide the layer, followed by **Show All** and **Hide All** layers.

This control is similar to the hide/show feature of L-Edit. For more information on this feature, refer to Showing and Hiding Layers on page 1-221 of the *L-Edit User Guide*. For more information on setting up colors, refer to Color Parameters on page 1-103 of the *L-Edit User Guide*.

## Status Bar

The **Status Bar**, located at the bottom of the L-Edit window, displays context-sensitive information on items in the interface. The status bar contains two panes. The right pane usually displays the L-Edit mode for layout views.

The left pane displays the status of the **3D Model View** as indicated in the following table.

<i>Action</i>	<i>Description</i>
The pointer is in the <b>3D Model View</b> palette	The name of the identified layer. If a layer is generated, this will be the Boolean formula for that layer.
A menu item is highlighted	A list of the menu's commands.

<i>Action</i>	<i>Description</i>
The pointer is in the tool bar	The function of the identified tool.
All other times	<b>Ready.</b>


The status bar may be displayed or not. To **Show** or **Hide** a status bar, select **View > Status Bars**.



Figure 91: **Status Bars** dialog

The checked status bars will appear as part of the viewer interface. Uncheck the bars you do not wish to see. They will immediately disappear. Click **Close** to exit the dialog.

## Viewing a Cross-section

From an active **3D Model View** window, clicking the  toolbar button snaps the 3D model to the top view and invokes cross-section viewing, bringing up the following dialog:

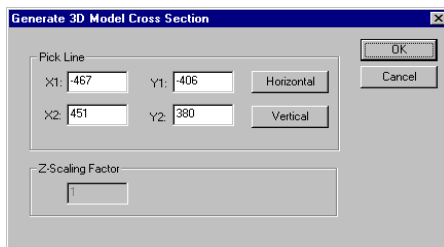


Figure 92: **Generate 3D Model Cross-Section** dialog

All cross-sections are performed perpendicularly to the surface of the wafer. The cross-section line is the intersection of the cross-section plane and the **X-Y** plane of the 3D model. The endpoints of the cross-section line are the **X-Y** pairs in the parameter list below.

**(X1, Y1), (X2, Y2)**

Coordinates of the cross-section line.

**Z-Scaling Factor**

The ratio of the height (Z) to the horizontal or vertical baseline (Not available in Version 3).



The orientation of the cross-section line along the width or length of the substrate may be set with the **Horizontal** or **Vertical** buttons. Choosing the **Horizontal** button will set **Y2** to the same value as **Y1**. Choosing the **Vertical** button will set **X2** to the same value as **X1**.

The cross-section line will appear in the 3D model window; the cross-section view itself will appear in a new window.

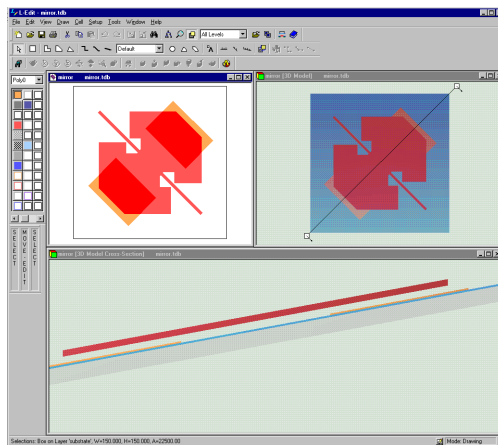



Figure 93: Performing the cross-section of a torsional mirror

In Figure 24, the various steps of the cross-section of a torsional mirror are displayed (clockwise from the upper left): the 3D model, the top view with the cross-section line, and the cross-section view. Note that the windows are tiled.

The cross-section line may be graphically modified by dragging its ends. As the cross-section line is moved, the cross-section view will automatically be updated. Simultaneous views of different cross-sections are not possible, since each time the cross-section line is moved, the cross-section is redrawn in the 3D cross-section window.

The title of the 3D cross-section view reads  **CellName [3D Model Cross-Section] Filename**. Multiple cross-section views of a single 3D model view cannot be made.

## Deleting 3D Models

To delete a 3D model, select **3D Tools > Delete 3D Model** in the **MEMS Pro Toolbar**. The following dialog will appear.

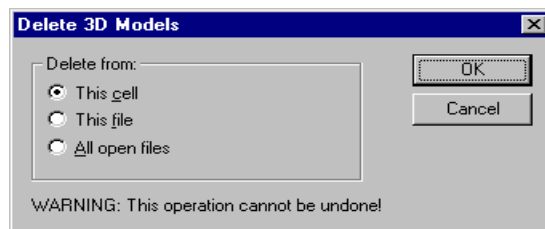


Figure 94: **Delete 3D Models** dialog

Select **This cell** to remove the 3D model in the active cell. Select **This file** to remove all 3D models in the active file. Select **All open files** to remove all 3D models in all open files.

The **Delete 3D Models** dialog can also be accessed from the context-sensitive menu of the **Design Navigator**. From an active layout window, select **View > Design Navigator**. To activate the context-sensitive menu, right-click on a cell. Select the **Delete 3D Model** command to invoke the dialog above.

**Warning**

---

*This operation cannot be undone.*

---

## Exporting 3D Models

3D models may be exported to **SAT** or **ANF** formats. The **Export 3D Model** option may be reached from L-Edit through the **3D Tools** menu of the **MEMS Pro Toolbar**. Once accessed, the **Export 3D Model** dialog will request the destination and format of your output file.

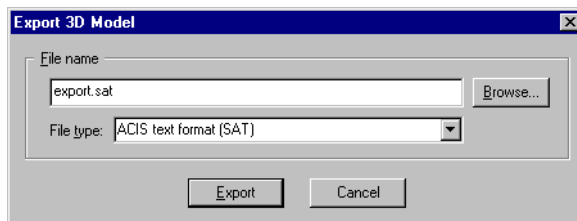


Figure 95: **Export 3D Model** dialog

Click the **Export** button to export a file with the name and type you selected.

**Export 3D Model** can also be accessed through the context-sensitive menu within the **Design Navigator**. The **Design Navigator** can be accessed from L-Edit

through the **View** menu. Right-click on the cell of interest to access the **Export 3D Model** command.

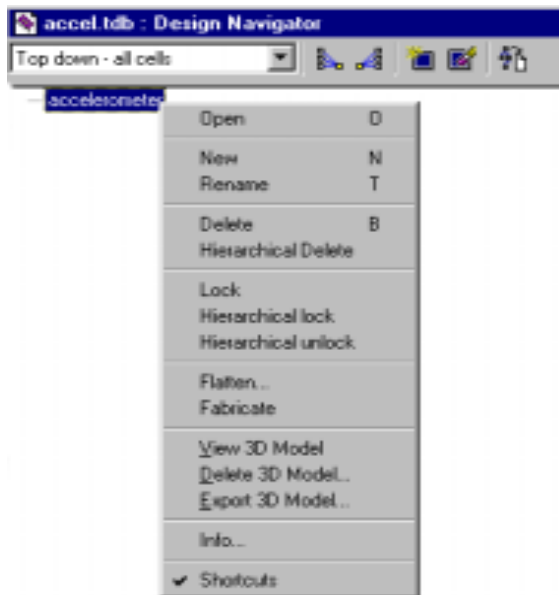


Figure 96: Accessing the **Export 3D Model** option via the context-sensitive menu

For more information, refer to Design Navigator on page 1-353 of the *L-Edit User Guide*.

If you plan to view the model in another graphics program, then exporting the file in **SAT** format, a general interchange format for solid models, is sufficient. If you plan to use ANSYS to analyze your model, the decision to export in **SAT** or **ANF** format will depend on your preferences and on your operating system.

ANSYS is able to import **ANF** files under any operating system, and it is always possible to write directly to **ANF** format from MEMS Pro. Under Windows NT and UNIX, MEMS Pro users also have the option of writing to **SAT** format.

Both MEMS Pro and ANSYS use an ANSYS module called *The ANSYS Connection Product for SAT* to convert **SAT** files to **ANF** format. You must have this connection module in your ANSYS directory both to export **ANF** files from MEMS Pro, and to read **SAT** files into ANSYS.

## Linking to ANSYS

Once you have successfully exported your model for use by ANSYS (see Exporting 3D Models on page 220), you are ready to invoke the program.

The direct link to the ANSYS program can be accessed by clicking the ANSYS button from the **3D Model Toolbar**. If the 3D Modeler cannot find the ANSYS executable, the following dialog will appear.

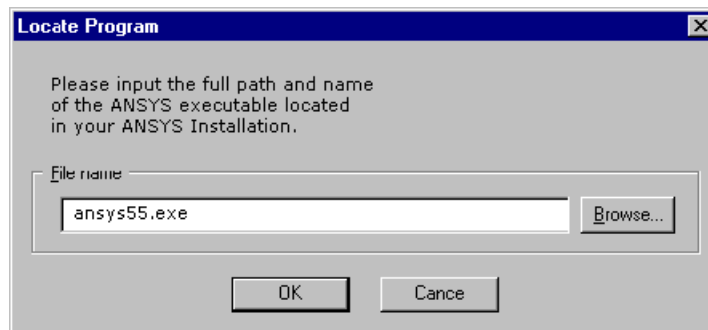


Figure 97: **Locate Program** dialog

Once you have located your ANSYS installation by browsing or typing the path to it, click **OK**. L-Edit will make a record of this location in the Windows



registry. You will not see this query again unless you move the ANSYS executable.



## Editing the Process Definition

Process definitions may be imported, exported, and edited from the **Process Definition** dialog. This dialog can be accessed from the **MEMS Pro Toolbar** selecting **3D Tools > Edit Process Definition**. Note that if the design file does not contain a process definition, the dialog will appear empty, as shown below.

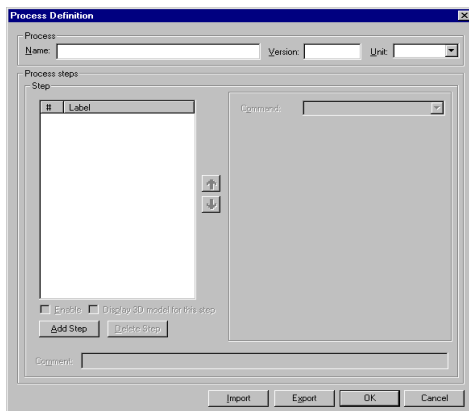


Figure 98: **Process Definition** dialog

To construct a 3D model, process definition information must either already be present in the design, entered manually through the **Process Definition** dialog, or imported from a process definition (**.pdt**) file.

## Importing the Process Definition

The process definition may be imported by clicking **Import** in the **Process Definition** dialog. An **Open** dialog will appear, as shown below.

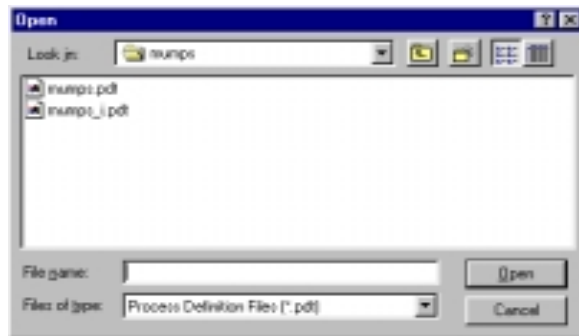


Figure 99: **Open** dialog

After locating and selecting the process definition file, click the **Open** button to populate the **Process Steps** list in the **Process Definition** dialog. Click **OK** again to import the process information into the 3D Modeler.

When the file is saved, the process definition will be attached to the **.tdb** file. The next time the layout is opened, the process information will be available to construct a 3D model; it will not have to be re-entered.

The **Process Definition** dialog may be used to **Add Steps**, **Delete Steps**, and to edit the parameters of an existing **Process Step**.

A MUMPS process definition has been imported into the dialog below. The commands correspond to the example used in the chapter on Process Definition on page 352.

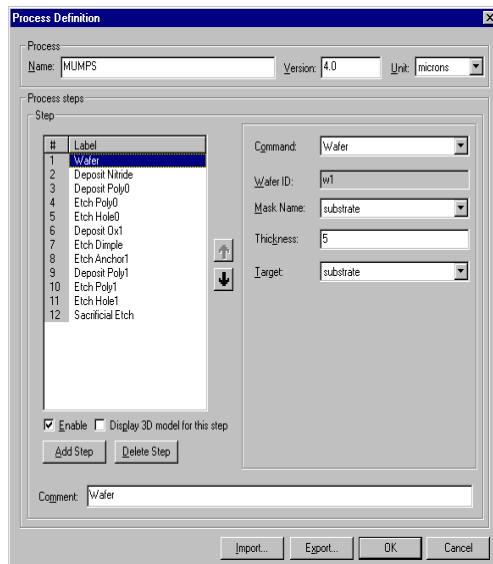


Figure 100: **Process Definition** dialog box

**Process** identification information appears at the top of the dialog. A **Process Step** may be edited in the body of the dialog. A **Comment** for each **Process Step** may be entered at the base of the dialog.

## Process Identification

At the top of the process definition dialog, you may enter identifying information.

<b>Name</b>	The name of the process definition.
<b>Version</b>	The version string of the process definition.
<b>Units</b>	Browse options for units are <b>microns</b> , <b>millimeters</b> , <b>centimeters</b> , <b>mils</b> , <b>inches</b> , <b>lambda</b> , and <b>other</b> .

## Editing the Process Steps List

The process definition editor has several useful features. Steps may be added, removed, rearranged or disabled (that is, commented out). 3D models of intermediate processing steps can be displayed at points set in the **Process Steps** list.

Each **Process Step** is identified by an order number and by a label. Below the **Process Steps** list are two check boxes: **Enable** and **Display 3D model for this**

**step.** These check-boxes select options to be applied to the selected **Process Step**.



### *Enable*

**Process Steps** are enabled by default. If you would like to see the 3D model that is created by omitting a given step, highlight the step and uncheck the **Enable** box. The disabled step will appear gray in the **Process Steps** list.

### *Display 3D model for this step*

By default, the entire fabrication process is emulated to produce a 3D model. Intermediate models can be displayed by checking the **Display 3D model for this step** box. When the 3D Modeler begins a checked step, a new window will open to display the model as it exists at the conclusion of that step. A separate window, titled **cellname [3D Model Step #] filename**, will open for each step marked by **Display 3D model for this step**.

### *Move Step*

To the right of the **Process Steps** list are two arrows. These arrows allow you to move a selected (highlighted) **Process Step** up  or down  within the **Process Steps** list.

## *Add Step*

The **Add Step** button will insert a step below the currently selected step, and label it **New Step #**. The default step type is **Deposit**. Once the new step has been added, you can redefine the **Command** in the editing area to the right of the **Process Steps** list. Commands available in MEMS Pro Version 3 include **Deposit**, **Etch**, **Wafer**, and **MechanicalPolish**.

## *Delete Step*

You may delete steps from the process definition by selecting the step and clicking the **Delete Step** button.

# Editing Individual Process Steps

All **Process Steps** have three parameters in common: the **WaferID**, **Label**, and **Comment**.

### **WaferID**

This parameter identifies the wafer you will be working on. MEMS Pro Version 3 supports just one **Wafer**, so this value is set to a default value of **w1** and cannot be edited. Future versions of the software will support multiple wafers and user-assigned names.

<b>Label</b>	This string appears in the progress dialog while the step is interpreted during 3D model generation. Short, descriptive terms are best for labels.
<b>Comment</b>	A note describing each <b>Process Step</b> or <b>Command</b> in more detail may be entered here.

The first **Process Step** is automatically selected in the **Process Steps** list that appears on the left side of the **Process Definition** dialog when the process definition file is opened. It is usually the **Wafer** step.

## Wafer

**Wafer** is selected in the **Process Steps** list on the left side of the **Process Definition** dialog below. **Wafer** also appears in the **Command** browse box on the right side of the dialog. The **WaferID** appears below it. Since MEMS Pro Version 3 only supports one **Wafer**, the **WaferID** is assigned automatically. It appears in gray and cannot be edited.



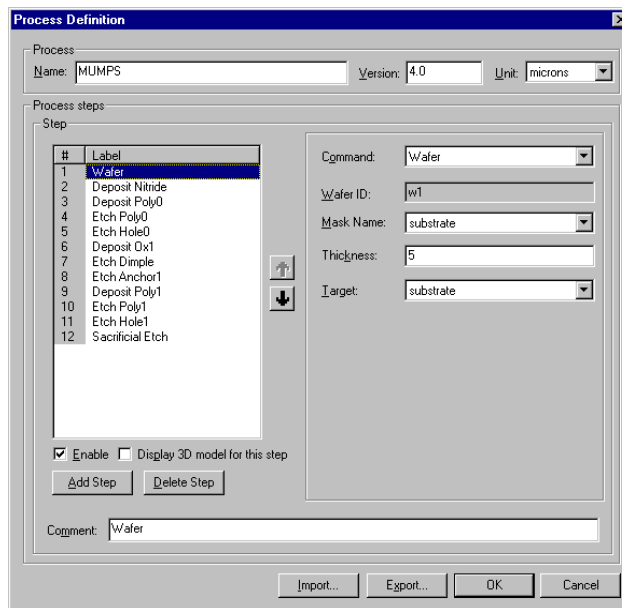


Figure 101: Characteristics of the Wafer step

Other parameters for the **Wafer** command are **MaskName**, **Thickness**, and **Target**.

**MaskName** **MaskName** choices include the list of layers in the design file. The geometry drawn on this layer defines the extent of the wafer. The boundary of the mask is usually defined by a box, but any drawing object may be used, including circles and curved polygons. Multiple objects that are not touching can also be drawn on the **MaskName** layer to define the **Wafer** extent. If there are objects drawn on other layers whose boundaries extend beyond the drawn **Wafer** extent, those objects will be truncated as the 3D model is built. If no closed curve is drawn on the **MaskName** layer, its extent will be set to 110% of the minimum bounding box of the layout on all other masks.

**Thickness** Any positive value is acceptable for the vertical height of the **Wafer**.

**Target** **Target** choices include the list of layers in the design file. This parameter specifies the 3D model rendering characteristics of the **Wafer**. **Target** and **MaskName** are typically set to the same layer. For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 186.

## Deposit

If the process **Command** is set to **Deposit**, a new set of parameters will appear to the right of the **Process Steps** list. These parameters are **DepositType**, **Face**, **LayerName**, **Thickness**, **Scf**, and **Target**.

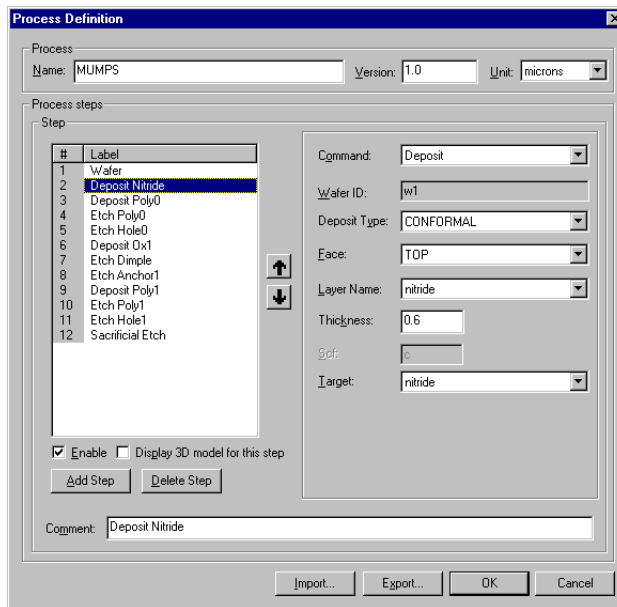


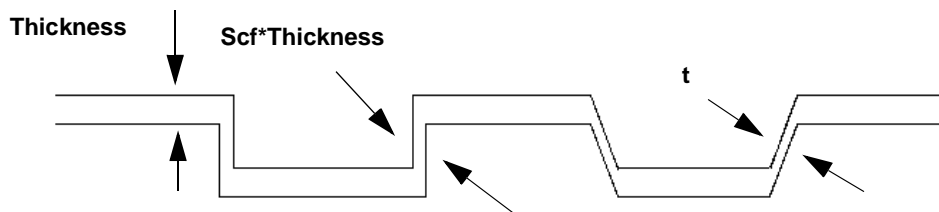
Figure 102: Characteristics of the Deposit Nitride step

The three possible values of **DepositType** are **CONFORMAL**, **SNOWFALL**, and **FILL**.

**CONFORMAL** deposit adds a layer that follows the contour of the processed wafer. **SNOWFALL** covers only those surfaces that are not shadowed by other surfaces on the wafer. **FILL** is a maskless **Process Step** that makes the surface of the wafer a plane. Each **DepositType** has unique parameter requirements.

*DepositType = CONFORMAL*

A **CONFORMAL** deposit is illustrated below.



Parameters for **CONFORMAL** deposits are **Face**, **LayerName**, **Thickness**, **Scf**, and **Target**.

<b>Face</b>	Parameter options include <b>TOP</b> , <b>BOT</b> (for bottom), and <b>TOPBOT</b> (for both top and bottom). <b>Face</b> identifies the side(s) of the wafer to receive the deposit.
<b>LayerName</b>	Parameter choices include the list of layers for the design. <b>LayerName</b> identifies the layer to be deposited; it is typically set to the same value as <b>Target</b> .
<b>Thickness</b>	Any positive number can be entered for the vertical dimension of the <b>CONFORMAL</b> deposit. This thickness is deposited on the side(s) specified by the <b>Face</b> parameter.
<b>Scf</b>	The <b>Scf</b> parameter is not supported in MEMS Pro Version 3 and therefore it may not be edited. Its value is assumed to be <b>1.0</b> or <b>c</b> for this release.

The **Scf** (*Sidewall coverage factor*) is the height of the material deposited on vertical sidewalls divided by the **Thickness** of the material deposited on horizontal surfaces of a **CONFORMAL** deposit. The material coverage **t** on walls at intermediate angles depends on the angle of inclination of the sidewall according to the relationship described in the section on Thickness and Scf on page 368. Entries for **Scf** can be a decimal number between **0** and **1**, or the letter **c**. An **Scf** of **c** is equivalent to an **Scf** of **1.0**, which is a

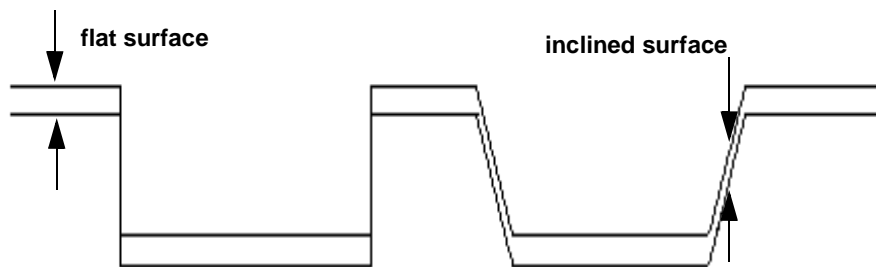
completely conformal deposit, that is, a deposit with uniform thickness along the wafer contour.

**Target** Parameter choices include the list of layers in the design file. **Target** specifies the 3D model rendering characteristics of the deposited layer. **Target** and **LayerName** are typically set to the same value.

For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 186.

*DepositType = SNOWFALL*

**SNOWFALL** deposits no material on vertical and shadowed surfaces, as shown below. Horizontal surfaces have the deepest coverage. Inclined surfaces have an intermediate amount of material deposited upon them.



Possible parameters for **SNOWFALL** deposits are **Face**, **LayerName**, **Thickness**, and **Target**.

**Face**

Parameter options include **TOP**, **BOT** (for bottom) and **TOPBOT** (for both top and bottom). **Face** identifies the side(s) of the wafer to receive the deposit.

**LayerName**

Parameter choices include the list of layers for the design. **LayerName** identifies the layer to be deposited; it is often set to the same value as **Target**.

**Thickness**

Any positive decimal number may be entered for the vertical dimension of the **SNOWFALL** deposit. This thickness is deposited on the side(s) specified by the **Face** parameter.

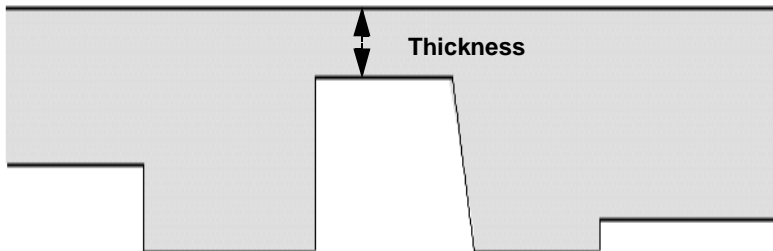
**Target**

**Target** choices include the list of layers in the design file. **Target** specifies the 3D model rendering characteristics of the deposited layer. It is typically set to the same value as **LayerName**.

For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 186.

***DepositType = FILL***

As illustrated below, the **Thickness** of **FILL** is set from the highest point of the model at that step for the **TOP Face**.





Possible parameters for **FILL** deposits are **Face**, **LayerName**, **Thickness**, and **Target**.

<b>Face</b>	Parameter options include <b>TOP</b> , <b>BOT</b> (for bottom) and <b>TOPBOT</b> (for both top and bottom). <b>Face</b> identifies the side(s) of the wafer to be filled.
<b>LayerName</b>	Parameter choices include the list of layers for the design. <b>LayerName</b> identifies the layer to be deposited; it is often set to the same value as <b>Target</b> .
<b>Thickness</b>	The vertical dimension of the <b>FILL</b> deposit as measured from the highest point on the <b>Wafer</b> up for the <b>TOP</b> face, or from the lowest point of the <b>Wafer</b> down for the <b>BOT</b> face (See the figure on page 377). <b>Thickness</b> may be any positive decimal number. The material is deposited on the side(s) specified by the <b>Face</b> parameter.
<b>Target</b>	Parameter choices include the list of layers in the design file. <b>Target</b> specifies the 3D model rendering characteristics of the filled layer. It is typically set to the same value as <b>LayerName</b> .

For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 186.

## Etch

**Etch** is used to sculpt the terrain of the **Wafer**. If the process **Command** is set to **Etch**, new parameters will appear on the right side of the **Process Steps** list. The specific parameters required to define this step depend on the selected combination of **EtchType** and **EtchMask**.

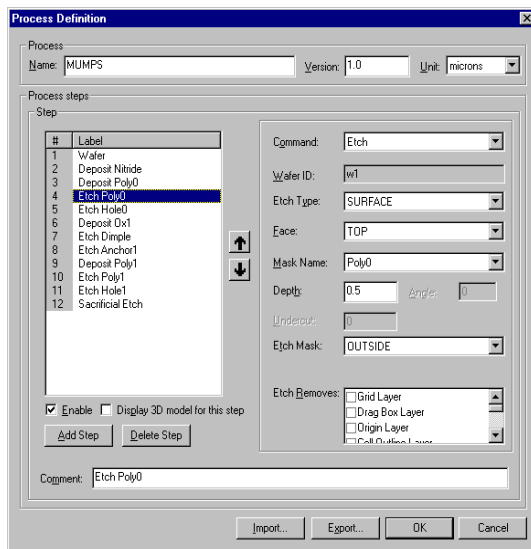


Figure 103: **Process Definition** dialog box

Possible **EtchTypes** are **SURFACE**, **BULK**, and **SACRIFICIAL**. Possible **EtchMasks** are **INSIDE** and **OUTSIDE**.

**SURFACE** etches remove material that has been deposited during previous steps. **BULK** etches remove parts of the **Wafer**. A **SACRIFICIAL** etch completely removes all bodies on the **EtchRemoves** layers. It does not require masking, and therefore there is no setting for **EtchMask** or **MaskName** for a **SACRIFICIAL** etch.

The orientation of the **Wafer** must be taken into account when setting these parameters.

### *Orientation Considerations*

The **Face** to be etched may be the top face (**TOP**), the bottom face (**BOT**), or both faces simultaneously (**TOPBOT**). If you are designing masks for processing on both faces of the wafer, you must be careful of the orientation of the masks. As Alan Nutt of Kodak Research Laboratories points out, to ensure correct alignment (as drawn in layout) of the masks designed for processing on the bottom of the wafer with the masks designed for processing on the top of the wafer, the former must be flipped horizontally (i.e., left-right reversed). You may be required to perform the reversal yourself or have the mask maker perform it. Please consult your mask maker for further information.

Another consideration for **SURFACE** etch is whether the mask setting is *inclusive* or *exclusive*. **EtchMask** may be set to either **INSIDE** (inclusive) or **OUTSIDE** (exclusive).

For **EtchMask = INSIDE**, areas beneath the mask layer geometry are removed (generally used for insulator masks). For **EtchMask = OUTSIDE**, areas beneath the mask layer geometry are protected (generally used for conductor masks). Below, identical masks with different **EtchMask** settings illustrate this effect:

#### **EtchMask = INSIDE**



#### **EtchMask = OUTSIDE**

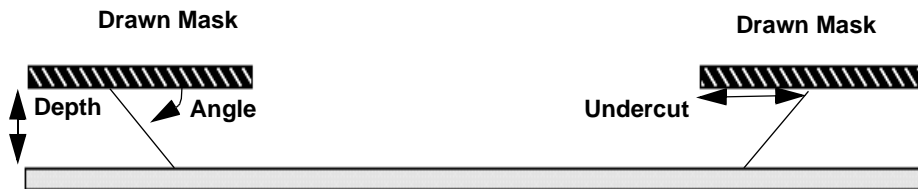


**SURFACE**, **BULK**, and **SACRIFICIAL** etches are described below.

## *EtchType = SURFACE*

The **SURFACE** etch removes layers specified in the **EtchRemoves** parameter.

Parameters for **SURFACE** etches include **EtchType**, **Face**, **MaskName**, **Depth**, **Angle**, **Undercut**, **EtchMask**, and **EtchRemoves**. In the diagram below, the parameter **EtchMask** is set to **OUTSIDE**.



**Face** Parameter options include **TOP**, **BOT** (for bottom) and **TOPBOT** (for both top and bottom). **Face** identifies the side(s) of the wafer to be etched.

**MaskName** Parameter options include the list of layers in the design. The geometry on this mask defines the area to be etched or excluded from etching.

<b>Depth</b>	<b>Depth</b> of material to be etched. Only the layers that are specified in the <b>EtchRemoves</b> parameter will be removed. For example, if the <b>Depth</b> is greater than the <b>Thickness</b> of the layer etched, the layer underneath will not be affected. Any positive value can be entered for <b>Depth</b> .
<b>Angle</b>	The <b>Angle</b> parameter is not supported in MEMS Pro Version 3 and therefore it may not be edited. Its value is assumed to be <b>90.0°</b> for this release.

Etch **Angle** is the angle of the sidewalls achieved by the etch.

<b>Undercut</b>	The <b>Undercut</b> parameter is not supported in MEMS Pro Version 3 for <b>SURFACE</b> etches and therefore may not be edited. Its value is assumed to be <b>0</b> for this release.
-----------------	---

For **EtchMask = INSIDE**, **Undercut** is the distance the etch front will extend over the drawn mask edge. For **EtchMask = OUTSIDE**, **Undercut** is the distance the etch front will intrude beneath the drawn mask edge.

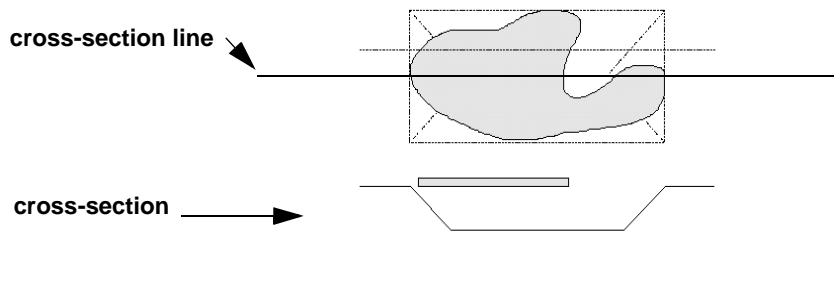
**Undercut = 0** is a sharply defined cut, aligned to the mask edge for both

cases.

- EtchMask** Parameter options are **INSIDE** or **OUTSIDE**. This parameter sets the mask orientation, that is whether the material to be removed is **INSIDE** or **OUTSIDE** of the drawn layout.
- EtchRemoves** Parameter options include the list of layers in the design; these appear in a scrolling checklist. Click on the name(s) of the layer(s) in the **EtchRemoves** list to mark them with an **X**. Marked layers will be removed during this **Etch** step.

### *EtchType = BULK*

The **BULK** etch sketched below is of KOH or EDP on a silicon wafer of **100** crystal orientation. The pit is bound by the **111** plane, which is attacked at a much slower rate than all other crystallographic planes. The outline of the box is the minimum bounding box of the mask pattern. This etch assumes **EtchMask = INSIDE**. The etch is viewed from above the **TOP** face. A cross-section corresponding to the dashed line appears below.



**BULK** etch parameters include **EtchType**, **Face**, **MaskName**, **Depth**, **Angle**, and **Undercut**.

**Face** Parameter options include **TOP**, **BOT** (for bottom) and **TOPBOT** (for both top and bottom). **Face** identifies the side(s) of the wafer to be etched.

**Depth** Vertical dimension of the etch. Any positive decimal number may be entered for **Depth**. Only the layers identified by the **EtchRemoves** parameter will be attacked.



<b>Angle</b>	Etch <b>Angle</b> is the angle of the sidewalls achieved by the etch, and is given as a decimal value between 45.0° and 90.0°.
<b>Undercut</b>	<b>Undercut</b> may be any positive decimal number. It is the distance the etch front will extend over the mask edge. <b>Undercut = 0</b> is a sharply defined cut, aligned to the mask edge.

### *EtchType = SACRIFICIAL*

A **SACRIFICIAL** etch completely removes all bodies on the **EtchRemoves** layers. This etch does not require masking and therefore has no setting for **EtchMask** or **MaskName**.

**SACRIFICIAL** etch parameters are **Face** and **EtchRemoves**.

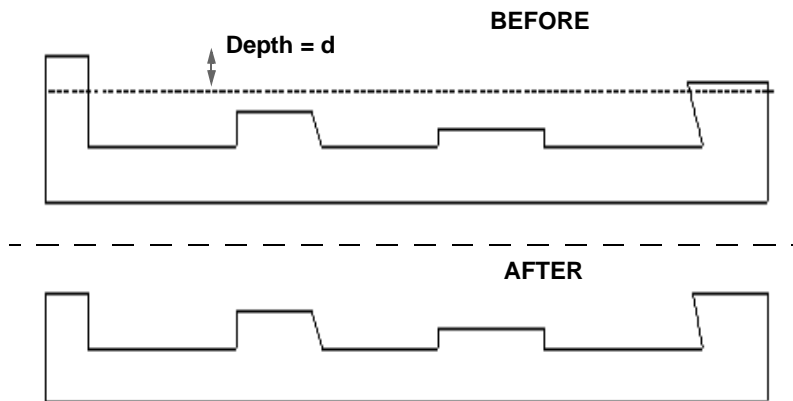
<b>Face</b>	Parameter options include <b>TOP</b> , <b>BOT</b> (for bottom) and <b>TOPBOT</b> (for both top and bottom). <b>Face</b> identifies the side(s) of the wafer to be etched.
<b>EtchRemoves</b>	Parameter options include the list of layers in the design; these appear in a scrolling checklist. Click the name(s) of the layer(s) in the <b>EtchRemoves</b> list to mark them with an <b>X</b> . Marked layers will be removed during this <b>Etch</b> step.

## MechanicalPolish

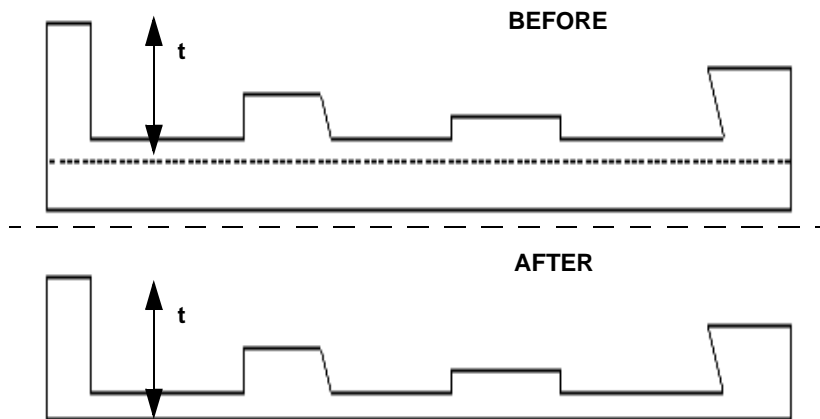
**MechanicalPolish** truncates the specified **Depth** off the top or bottom of the entire wafer, regardless of material type.

The effects of **MechanicalPolish** can be specified by either a **Depth** or a **Thickness**, *but not both*. When a **Depth** is specified, that **Depth** is truncated off the face of the wafer. When a **Thickness** is specified, that **Thickness** remains after polishing.

The drawing below gives the profile of a wafer before and after **MechanicalPolish**. The depth **d** has been sliced off the top of the wafer.



In the drawing below, the **MechanicalPolish** command has sliced material from the bottom of the **Wafer** and left **Thickness = t**.



**Face** Parameter options include **TOP** and **BOT** (for bottom). **Face** identifies the side of the wafer to be etched. Note that only one side of the wafer may be mechanically polished at a time. **TOPBOT** is not an available option for this step.

**Depth** **Depth** may be any positive decimal number. It is the vertical measure of the material *removed*, measured from the highest point of the **Wafer** for the **TOP** side, or from the lowest point of the **Wafer** for the **BOT** side.

**Thickness**

**Thickness** may be any positive decimal number. It is the vertical measure of the material that *remains* after the polish. It is measured from the lowest point of the **Wafer** for the **TOP** side and from the highest point of the **Wafer** for the **BOT** side.

## 3D Modeler Error Checks

The 3D Modeler performs several checks before presenting or generating the 3D model for view. These checks are the following:

- Is the 3D model out-of-date?
- Does the process definition exist?
- Are there derived layers in the process definition?
- Do all the required mask layers exist?
- Are there any visible wires or self-intersecting/ambiguous polygons in the mask layout?

These checks are described on the following pages.

### *Checking if the 3D Model is Out-of-Date*

An existing 3D model is made obsolete if the process definition or layout used to generate it has been altered. If a 3D model is out-of-date, a warning dialog will appear that states the situation (**3D Model Out-Of-Date**) and what changes have occurred since the model was last generated.

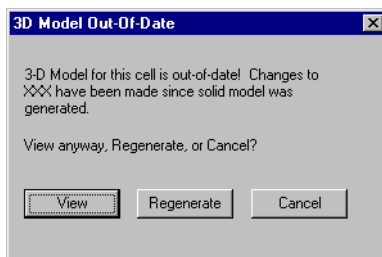


Figure 104: **3D Model Out-Of-Date** dialog

Select **View** to display the existing (outdated) 3D model. Select **Regenerate** to replace the existing 3D model. Select **Cancel** to quit the operation.

### *Checking if a Process Definition is used*

A 3D model cannot be generated without a process definition. Click **OK** to return to the layout view and add a process definition by accessing the **Edit Process Definition** dialog.

## *Checking for Process with Derived Layers*

If the process file refers to a derived layer, a note reminds you to use the L-Edit command **Generate Layers** before continuing. Click **Yes** to proceed with the generation of the 3D model. Click **No** to abort the operation.

### **Note**

---

For more information on *generating layers and derived layers*, see Introduction to Generated Layers on page 1-403 of the *L-Edit User Guide*.

---

## *Checking for the Existence of all Required Layers*

All layers specified in the process definition file must be present in the layer setup. If any of the layers referred to in the process definition does not exist in the layer setup, a warning is issued specifying the missing layers. Click **OK** to return to the layout view. Once in layout view, the missing layers can be added.

## *Checking for Wires or Self-Intersecting Polygons*

Wires and self-intersecting polygons are not currently supported by the 3D Modeler. If these objects exist in the layout a warning is issued that the objects will be ignored. The warning will list the cells that contain the unsupported objects. Click **OK** to ignore the unsupported objects and proceed with model generation.

**Note**

---

For more information, see Polygons and Wires on page 1-248 of the *L-Edit User Guide*.

---



# 7 ANSYS Tutorial

---


u ■ Introduction	257
■ Reading the 3D Model in ANSYS	261
■ Setting Boundary Conditions	265
■ Meshing the Model	269
■ Running the Analysis	272
■ Displaying the Results	273
■ Computing the Spring Constant	276
■ Entering Models under Windows NT	277

## Introduction

Files created by the 3D Modeler can be read into the ANSYS program. Once the 3D model is entered, it can be analyzed using any ANSYS finite element module.

In this tutorial, you will perform a simple structural analysis of the spring mechanism on a lateral comb resonator. You will apply a small force on one end of the device model. ANSYS will compute the resulting deflection. You will then calculate the spring constant using Hooke's law.

### Launching L-Edit

- ☒ Launch L-Edit by double-clicking the **L-Edit** icon  located in the installation directory. A default file named **Layout1** should be visible in the work area.
- ☒ Close the **Layout1** file by selecting **File > Close**.

### Opening the File

- ☒ Use **File > Open** to open the file named **spring.tdb** in the **<install directory>\tutorial\ansys** directory.

The spring (see Figure 105) whose layout is shown below, was designed to be fabricated with the MCNC MUMPS technology. It is on the second polysilicon layer (Poly1) suspended over the ground plane. It is anchored to the ground plane formed on the first polysilicon layer (Poly0) at the small area on the lower center of the spring. There are three dimples at the top of the spring.

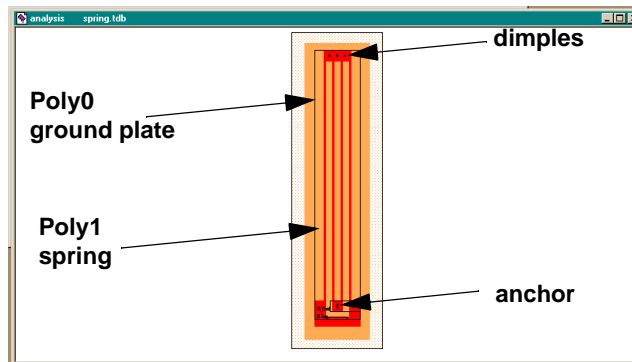


Figure 105: Viewing the spring

## Viewing the 3D Model

The **spring.tdb** file already contains a 3D model built from the layout and MUMPS manufacturing process. You can find out more about how models are constructed in the main tutorial section entitled Viewing a 3D Model on page 71.

- ☒ In the MEMS Pro Palette, choose **3D Tools > View 3D Model** to view the model.

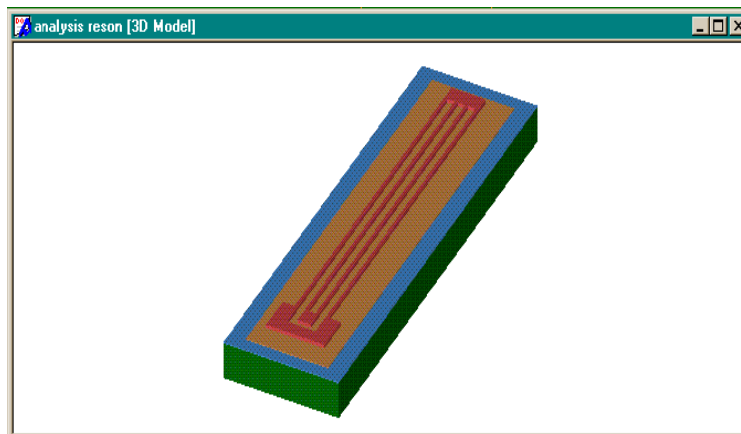


Figure 106: Viewing the 3D model of the spring

## Exporting the 3D Model

Two file formats used in MEMS Pro to describe 3D models are **ANF** (ANSYS Neutral Format) and **SAT** (Save As Text). MEMS Pro can write **SAT** files directly under all operating systems.

Under Windows 95, you must export your 3D model in **ANF** format to use with ANSYS. MEMS Pro uses the ANSYS connection module called *The ANSYS*

*Connection Product for SAT* to write the **ANF** file that describes your model. You must have this connection module installed in your ANSYS directory to accomplish this task.


Under Windows NT and UNIX, MEMS Pro users who wish to use ANSYS can export their files in either **ANF** or **SAT** format. If you choose to export to **ANF** format, MEMS Pro will invoke *The ANSYS Connection Product for SAT* as you export the file. If you choose to export your model from MEMS Pro in **SAT** format, ANSYS will invoke *The ANSYS Connection Product for SAT* as it reads the **SAT** file. In either case, you must have *The ANSYS Connection Product for SAT* installed in your ANSYS directory.

In this tutorial, you will export the 3D model as an **ANF** file for analysis in ANSYS.

- ☒ Choose **Tools > Export 3D Model**.
- ☒ Set the file type to **ANF**, and the file name to **spring.anf**, and click **Export**.

Refer to **<install directory>\ToAnsys\ansys.wri** for details on connecting MEMS Pro output to ANSYS input.

## Reading the 3D Model in ANSYS

- ☑ Launch ANSYS by clicking the **ANSYS** button  on the **3D Model View** toolbar.

The location of ANSYS depends on your individual system; you may have to browse your file system to find the ANSYS executable. The default location is **c:\ansys55\bin\Intel\ansysir.exe**.

- ☑ In the **ANSYS Utility** menu, choose **File > Read Input from** and browse for the **spring.anf** file.

## Viewing the 3D Model in ANSYS

Once the 3D model has been read, it may be viewed in several ways.

- ☑ In the **ANSYS Utility** menu, choose **Plot > Volumes** to show the edges of the 3D model. To view the model with shaded surfaces, from the **ANSYS Utility** menu, choose **PlotCtrls > Reset Plot Ctrls**, then choose **Plot > Volumes** again.
- ☑ In the **ANSYS Utility** menu, choose **PlotCtrls > Pan-Zoom-Rotate**.
- ☑ In the **Pan-Zoom-Rotate** menu, check the box labeled **Dynamic Mode**. The left mouse button now controls panning and the right mouse button controls rotation.

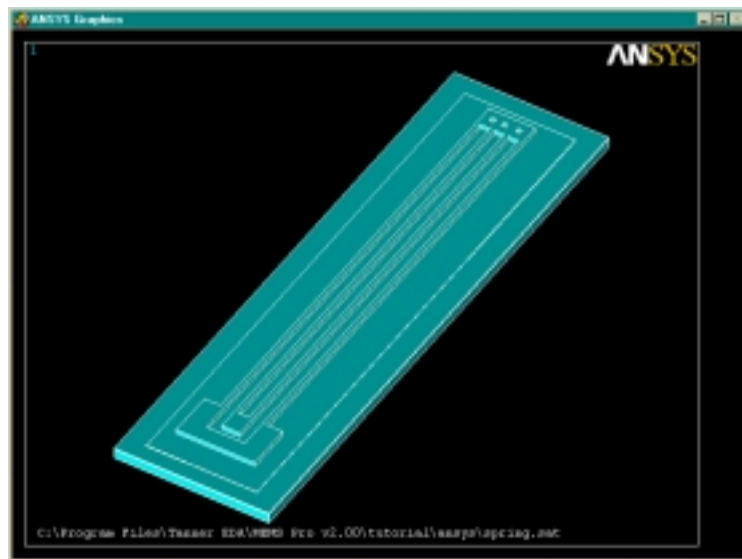


Figure 107: 3D model of the spring displayed in ANSYS

## Setting Material Properties

Our example assumes that the spring mechanism is made of polysilicon that has a Young's modulus of 150 GigaPascals and a Poisson's ratio of 0.2. The 3D model, however, is defined in microns and not meters. Instead of converting the 3D model to meters we will perform all the calculations in this tutorial in a system of units consisting of microns, kilograms, and seconds. In these units the Young's modulus has the value of  $1.5 \times 10^5$  (the Poisson's ratio is dimensionless, so it is unchanged).

You will now enter these material properties for polysilicon into your model.

- ☒ In the **ANSYS Main** menu, choose **Preprocessor > Material Props > Constant–Isotropic**.
- ☒ In the **Isotropic Material Properties** dialog, verify that the material number is set to **1** and click **OK**.
- ☒ In the **Isotropic Material Properties** dialog, enter **1.5e5** for Young's modulus (**EX**) and **0.2** for Poisson's ratio (**NUXY**).
- ☒ Click **OK**.



## Adding an Element Type

The element type specifies the mesh element shape. It must be declared for the finite element before boundary conditions are set.

- ☒ In the **ANSYS Main** menu, choose **Preprocessor > Element Type > Add/Edit/Delete**.
- ☒ In the **Element Types** dialog, click **Add**.
- ☒ In the **Library of Element Types** dialog, choose **Structural Solid** in the left box.
- ☒ In the right box, scroll to the **Tet 10node 92** entry. Select **Tet 10node 92**. Click **OK**.

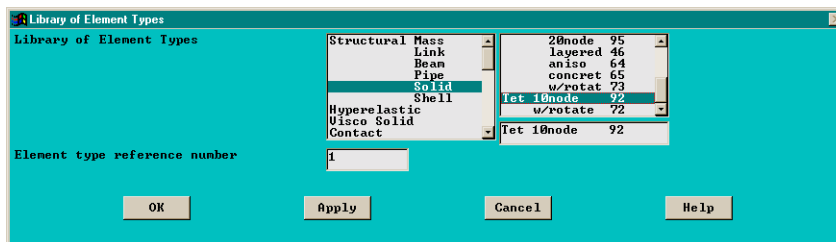


Figure 108: **Library of Element Types** dialog

- ☒ Click **Close** in the **Element Types** dialog.

## Setting Boundary Conditions

You will anchor the spring to the surface it shares with the layer below it (this is surface number 59 in the 3D model) and apply a small, leftward pointing force to the two keypoints (numbers 41 and 42) on the upper right corner of the model. This will cause the spring to bend slightly to the left.

First, anchor the spring.

- ☑ Picking the correct area will be easier if only the edges of the 3D model are displayed. Choose **Plot > Lines** from the **ANSYS Utility** menu and then **PlotCtrls > Pan–Zoom–Rotate**. Zoom in on the area near the anchor using either one of the zooming tools in the **Pan–Zoom–Rotate** menu. It will be helpful in picking the correct area to rotate the 3D model so that the viewing angle is not directly from above.
- ☑ In the **ANSYS Main** menu, choose **Preprocessor > Loads > Loads–Apply > Structural–Displacement > On Areas**. The **Apply U,ROT on Areas** picking menu will appear.
- ☑ Now hold the left mouse button down and drag the pointer around the display. Notice that different areas are highlighted while the pointer is over them. Drag the pointer over the anchor area until it is highlighted, as in Figure 109. Releasing the mouse button will pick this area (which should be number **59**). If you

accidentally select another area, you can click **Reset** in the **Apply U,ROT on Areas** picking menu to unselect it.

- ☑ Once the correct area (and only this area) is selected, click **OK** in the **Apply U,ROT on Areas** picking menu.

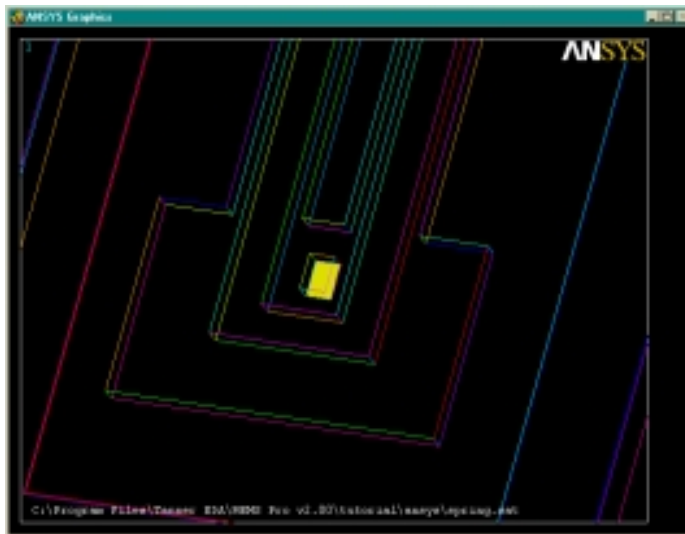


Figure 109: Selecting a particular area of the 3D model

- ☑ In the **Apply U,ROT on Areas** dialog, verify that **DOFs to be constrained** is **All DOF** and **Apply as** is **Constant value**. Type **0.0** for **Displacement value**. Click **OK**.

Next, locate the keypoints, where the testing force will be applied.

- ☑ Click **Fit** in the **Pan–Zoom–Rotate** menu. Now zoom in on the opposite end of the model, near the dimples.
- ☑ In the **ANSYS Main** menu, choose **Preprocessor > Loads > Loads–Apply > Structural–Force/Moment > On KeyPoints**. The **Apply F/M on KPs** picking menu will appear.

- ☑ Using the same technique as above, select the two keypoints on the upper right end of the spring. These should be numbers **41** and **42**. Refer to Figure 110 to check if you have selected the appropriate keypoints.



Figure 110: Selecting two keypoints on the spring

- ☑ Making sure that only these two keypoints are selected, click **OK** in the **Apply F/ M on KPs** picking menu.

- ☑ In the **Apply F/M on KPs** dialog, make sure **FX** is shown for the **Direction of force/mom** and **Apply as** is set to **Constant value**. For **Force/moment value** enter **-1.0**, i.e., the force on each node is pointing in the negative **X** direction and has a magnitude of one microNewton (remember our units are microns/kilograms/seconds). Thus, the total leftward pointing force on the spring is two microNewtons.

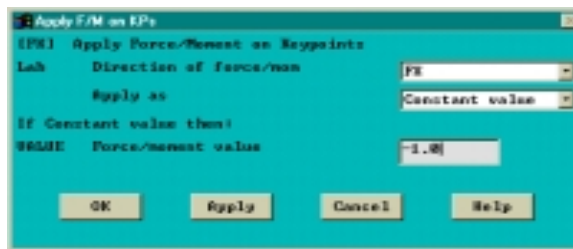


Figure 111: **Apply F/M on KPs** dialog

- ☑ Click **OK**.

## Meshing the Model

You are now ready to mesh the model.

- ☑ In the **ANSYS Main** menu, choose **Preprocessor > MeshTool**.
- ☑ In the **MeshTool** dialog, check the **SmartSize** box.
- ☑ Position the slider so that smartsizing is set to **8**.
- ☑ Verify that **Volumes** is selected for **Mesh**, that **Shape** is set to **Tet** and that **Mesher** is set to **Free**.
- ☑ Click **Mesh**.

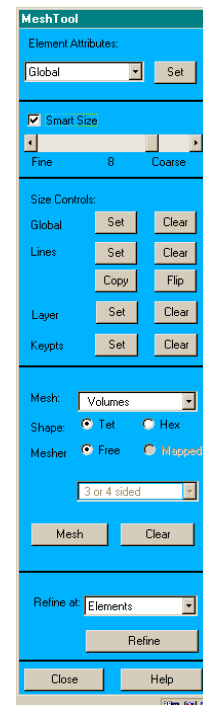


Figure 112: **Mesh Tool** dialog

- ☑ The **Mesh Volumes** picking menu will appear. Specify the volume for meshing by its number, rather than selecting it with the mouse.
- ☑ Select the spring by typing **4** in the **ANSYS Input** window, then press **Return**.
- ☑ Click **OK** in the **Mesh Volumes** picking menu.

The mesher will take a short time to mesh the spring. When the mesh is completed the elements are displayed. This display can be manipulated by the **Pan-Zoom-Rotate** menu in the same way as that of the 3D model.

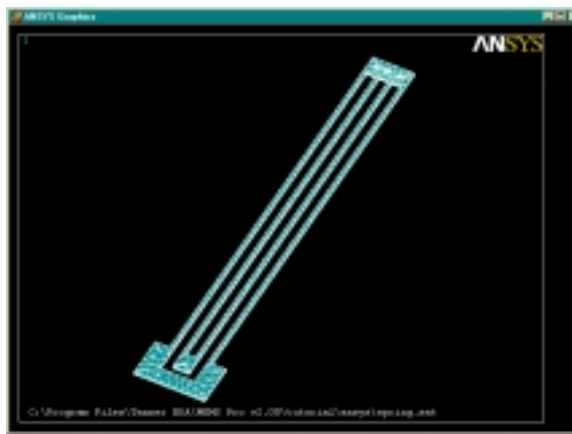


Figure 113: Meshed view of the spring



## Running the Analysis

Now that you have set material properties, applied boundary conditions, and meshed the model, you are ready to perform a linear structural analysis.

- ☑ In the **ANSYS Main** menu, choose **Solution > Solve—Current LS**.
- ☑ Take a moment to review the information in the **/STAT Command** window. Close this window by clicking the **Close** icon in the upper right corner.
- ☑ In the **Solve Current Load Step** dialog, click **OK**. Depending on CPU speed and memory allocation, the analysis may take several minutes.
- ☑ When the analysis is finished, an **Information** dialog will appear stating **Solution is done!**. Click **Close**.

## Displaying the Results

The results of the analysis are not immediately displayed. You must identify the results you want to display, and specify how you want them to be displayed.

- ☑ In the **ANSYS Main** menu, choose **General Postproc > Read Results—First Set**.
- ☑ In the **ANSYS Main** menu, choose **General Postproc > Plot Results > Contour Plot—Nodal Solution**.
- ☑ In the **Contour Nodal Solution Data** dialog, verify that **DOF solution** is selected in the left box and **Translation UX** in the right box for **Item to be contoured**. ANSYS will use the relative displacement in the **X** direction for the color scale on the contour plot.

- ☑ Choose **Def shape only** for **Items to be plotted**.

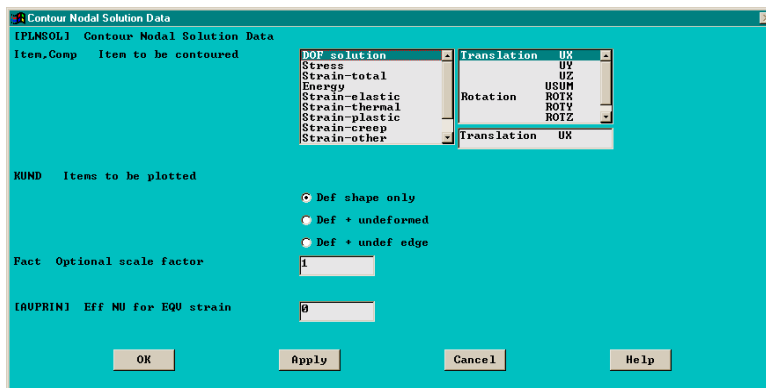


Figure 114: **Contour Nodal Solution Data** dialog

- ☑ Click **OK**.
- ☑ Choose **Front** from the **Pan–Zoom–Rotate** menu to view your results.
- ☑ Note that the deflection is not to scale. From the **ANSYS Utility** menu, choose **PlotCtrls > Style > Displacement Scaling**.
- ☑ In the **Displacement Display Scaling** dialog, select **1.0 (true scale)** as the **DMULT Displacement scale factor**.

- ☒ Click **OK**.

Now, the deflection is displayed to scale as in Figure 115.

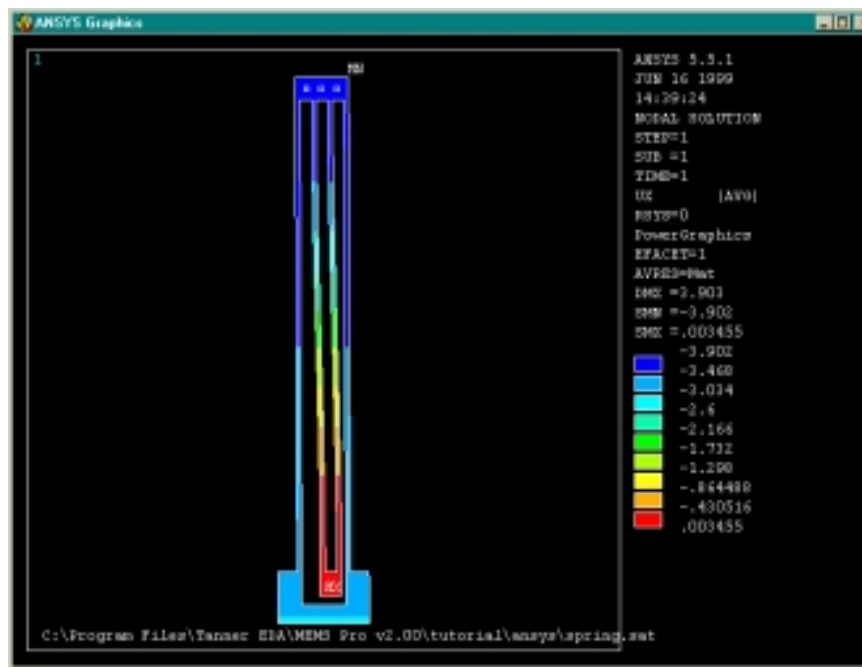


Figure 115: Deflection results

## Computing the Spring Constant

On the right side of the display window, notice that the maximum deflection is 3.9 microns. Thus, the spring constant for this device is approximately  $2 \text{ microNewtons} / 3.9 \text{ microns} = 0.5 \text{ Newtons/meter}$ . By varying the applied load it can be verified that the relationship between the load and the maximum deflection is linear.

## Entering Models under Windows NT

On Windows NT and UNIX systems, the **SAT** file may be read directly into ANSYS without the need to export it in **ANF** format. However, to use this capability you must have *The ANSYS Connection product for SAT* module installed.

- ☑ In the **ANSYS utility** menu, choose **File > Import > SAT** and browse for the ***spring.sat*** file.

You may also export your model from MEMS Pro in **ANF** format as described in Exporting the 3D Model on page 259.

## 8 ANSYS to Layout Generator

---

▪ Introduction	279
▪ 3-D to Layout Tools	281
▪ The Layout Generator Program	299
▪ Definition of a Technology File	303
▪ Limitations	315
▪ Tutorial	316
▪ Layout view of the mirror	333

# Introduction

The ANSYS 3D-Model to Layout Generator allows you to project an ANSYS database into a CIF file that can be read by almost all Electronic Design Automation tools.

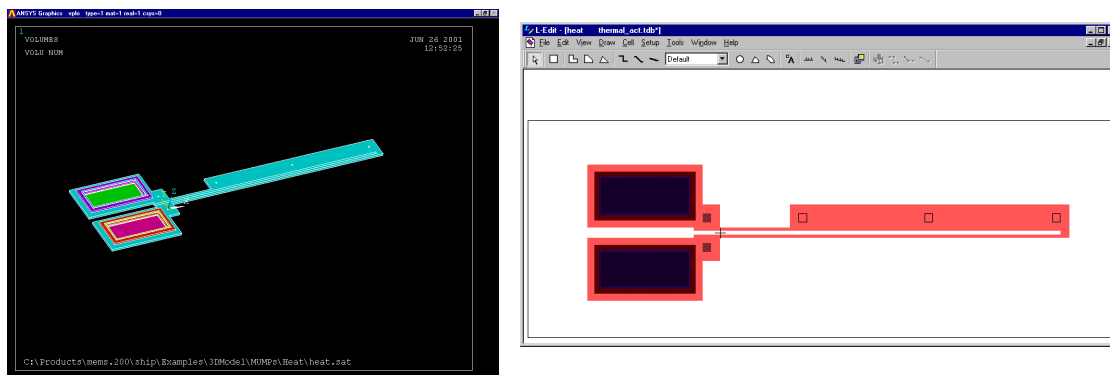


Figure 1: Horizontal heat actuator (3D and 2D views)

A palette of utilities, often used to create 3D structures, has been added to the MEMSCAP palette, in the **3D to Layout** menu. The utilities help you create keypoints, lines, arcs, areas and volumes by using the same method as the



standard ANSYS commands. But these utilities are necessary to the generation of a layout from a 3D model. Indeed, they associate to the created volumes a component name relating those volumes to the material it represents (i.e. to the layer). These component names are necessary to the 3D to Layout translator.

Options for editing volumes by moving them, subtracting or adding them have also been added to the palette. These modifications maintain the component name.

## 3-D to Layout Tools

### Overview

The **3-D to Layout** menu gathers frequently used commands, defined in several locations in ANSYS, with functions developed by MEMSCAP for the ANSYS to Layout translator.

- ☒ You can access the **3-D to Layout** menu through the ANSYS Main Menu by selecting **MEMSCAP Tools > 3-D to Layout** (Figure 2).

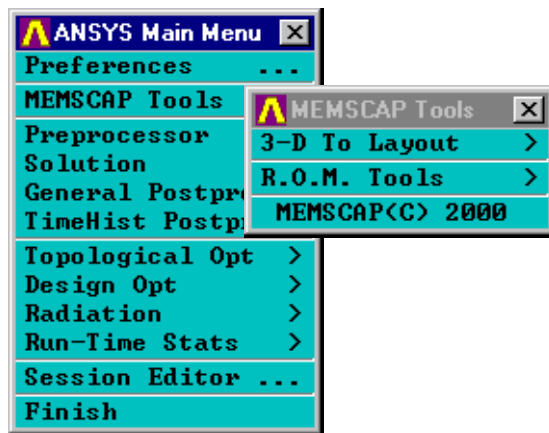


Figure 2: Accessing the **MEMSCAP Tools** menu

The **Prompt** dialog box appears (Figure 3).



Figure 3: Setting up the technology file name

- ☒ Enter the technology file name and press **OK** or **Return**.

This file defines the component names for the materials of the 3D model, and links them to mask layer names in the corresponding CIF file. See Component Names, for more details.

**Warning**

---

*The name of the technology file must be enclosed in single quotes.*

---

**Note**

---

ANSYS restricts variable name lengths to 8 characters. If you enter a component name of more than 8 characters, ANSYS only takes into account the first 8 characters. For example, if you enter “Bulketch1”, ANSYS reads “Bulketch”.

---

Once the techno name is entered, the **3-D To Layout** menu appears.

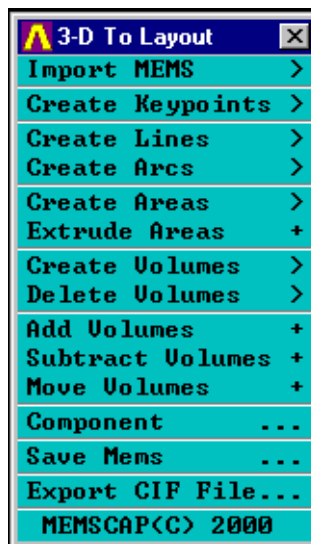


Figure 4: **3-D to Layout** menu

This menu (see Figure 4) helps you work with the 3D model before its translation to layout.

## Import Mems

- ☒ You can import a 3D model by clicking **Import MEMS** (Figure 5).

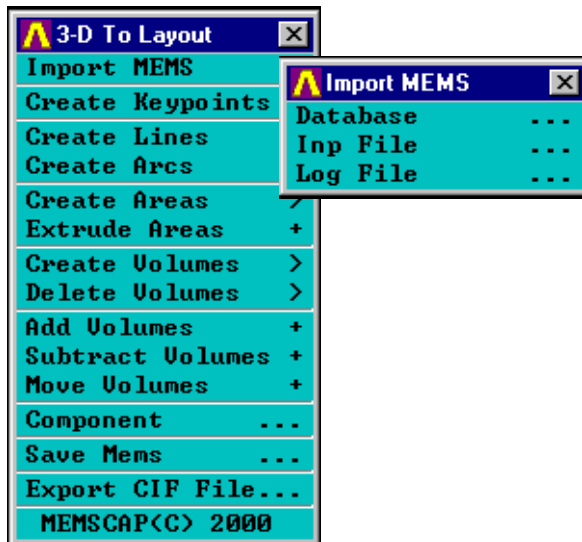


Figure 5: Reading the 3D Solid model

You can choose to read an input file (.inp), a log file (.log) or an ANSYS database file (.db). These commands are also available through the **Read Input From** (for ASCII files) or **Resume From** (for a database) buttons in the **ANSYS File** menu.

Any of these choices brings up the same following dialog box (Figure 6).

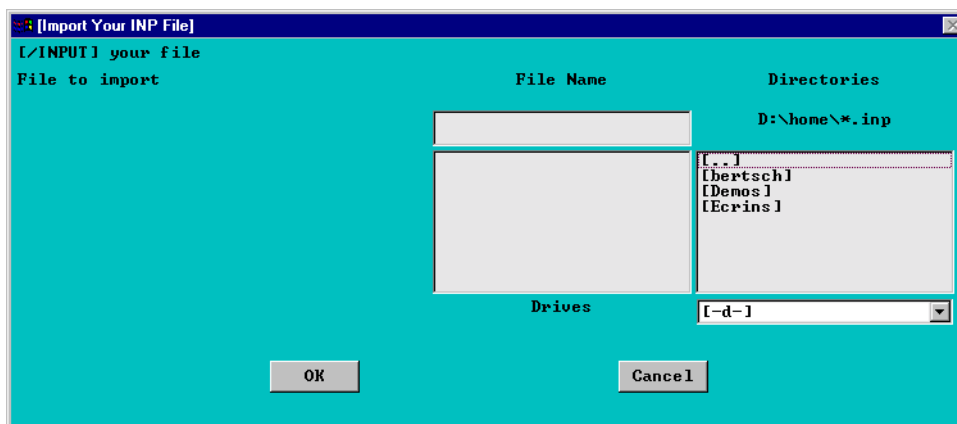


Figure 6: Reading a MEMS file in ANSYS

- ☒ Select the file you want to import.

## Creation of Volumes

You can create keypoints, lines, arcs and areas using the **3-D to Layout** palette. These commands are exactly the same as ANSYS commands but with some extra book-keeping related to the materials used.

- ☒ By clicking the **Create Volumes** button, you can create blocks, cylinders, prisms and volumes by areas (Figure 7).

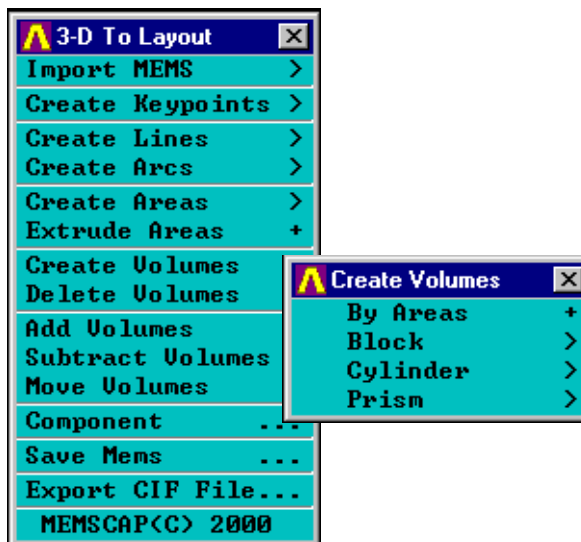




Figure 7: **Create Volumes** button

Once the volume is created, a new dialog box (see Figure 8) prompts you to select a component name for the volume. This component name is related to the name of the mask layer on which the 2D projection of the volume will reside after translation.

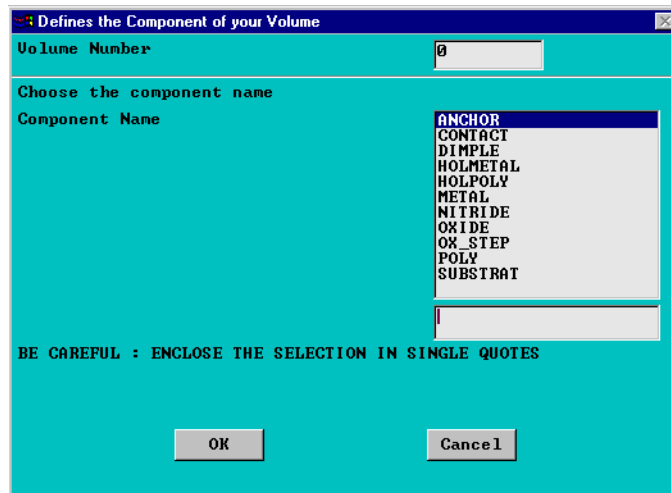


Figure 8: Attaching a component name to the created volume

If a volume has no component name, it cannot appear in the CIF file created by the ANSYS to Layout translator. If a volume has more than one component name, the program generates an error message.

If all the previously created volumes have valid component names, the **Volume Number** field is automatically filled with the next available number.

### **Warning**

---

*If your solid model contains volumes that are not attached to component names, the dialog box loads the smallest number of volumes. If you do not change the volume number, this volume is associated to the selected component name. The volume you first intended to create will not be attached to a component. You must fill the dialog box with the correct volume number.*

---

A scrolling list appears containing component names defined for the technology.

- ☒ Enter the component name in the appropriate field.
- ☒ Click **OK** to record the component name.

### **Warning**

---

*When entering a component name, you must enclose the name in single quotes.*

---

**Warning**

---

*In the layout, you cannot select hole layers because they correspond to nothing in ANSYS. For example, you cannot choose HOLMETAL or HOLPOLY1.*

---

The volume is the same color as the component name. You can choose another volume number if this one has no component.

If you make a mistake (forget the single quotes, for instance) or if you want to attach a volume to a component without creating a volume, you can click **Define Component** in the **3-D to Layout** palette and the same dialog box appears (Figure 8).

Once a component name is attached to a volume, you can modify it by deleting the component name associated with the selected volume and redefining it using the **Define Component** command in the **3-D to Layout** palette.

**Warning**

---

*If you remove a component name, this component is deleted in every volume containing this name.*

---

**Note**

---

ANSYS restricts variable name lengths to 8 characters. If you enter a component name including more than 8 characters, ANSYS only takes into account the first 8 characters. For example, if you enter "CONTACT\_POLY", ANSYS creates a

volume which component name is CONTACT\_. Since this name does not exist in the component list, and the volume will not appear in the layout.

---

## Deletion of Volumes

This is a standard ANSYS command.

- ☑ To delete volumes, click **Delete Volumes** in the **3-D to Layout** palette. Two options are available: **Volumes Only** and **Volumes & Below**. To delete the volumes and all the areas, lines and keypoints created with the volumes, click **Volumes & Below**.
- ☑ Then, select the volumes you want to delete by using the **Pick** radio button.

### **Warning**

---

*If you delete all the volumes attached to the same component, this component is also deleted and no longer appears in the component list.*

---

## Addition of Volumes

- ☒ To add volumes, click **Add Volumes** in the **3-D to Layout** palette.

After the addition of volumes, a dialog box appears indicating that ANSYS will add the areas of the new volumes. This is specific to MEMSCAP's implementation.

You can also subtract or move volumes by using the **Subtract Volumes** and **Move Volumes** options of the **3-D To Layout** menu.

These boolean operations delete the component of the added or subtracted volumes. After the addition or the subtraction, the dialog box (see Figure 8) appears in order to define a component name for the new volume.

If you move a volume with non-circular arcs and circular arcs, ANSYS loses the information about the arcs. For instance, if you create a sphere and move it, all the arcs of the sphere become straight lines.

## Component Names

When you click **Define Component**, the dialog box (Figure 8) appears to allow you to define the component name of the smallest volume which is not attached to a component.

Each component in ANSYS corresponds to the name of the layer in the resulting CIF file. Geometry on each CIF layer is made from projections of connected volumes of several components.

The relation between the name of the component in ANSYS and the name of the layer in the resulting layout for a specific example technology is shown in the following table.

For the technologies, some layers representing holes (HOLPOLY and HOLMETAL for example in the surfmic technology) are defined. Those layers are not used in the 3D models. They are in the components list, but you must not associate them with volumes.

**The surface micromachining process**

<b>COMPONENT NAME</b>	<b>LAYER NAME</b>
SUBSTRAT	-
POLY	poly
ANCHOR	anchor
DIMPLE	dimple
METAL	metal
CONTACT	contact
HOLPOLY	holpoly
HOLMETAL	holmetal

## Saving Mems

If you made some modifications to the 3D Solid Model, you can save them as a database (see Figure 9) with the **Save MEMS** option. It is exactly the same as the **Save as...** option in ANSYS.

- ☒ Select **MEMSCAP Tools > 3-D To Layout > Save MEMS** in the ANSYS Main Menu.

The **Save Database of your MEMS** dialog box appears (Figure 9).



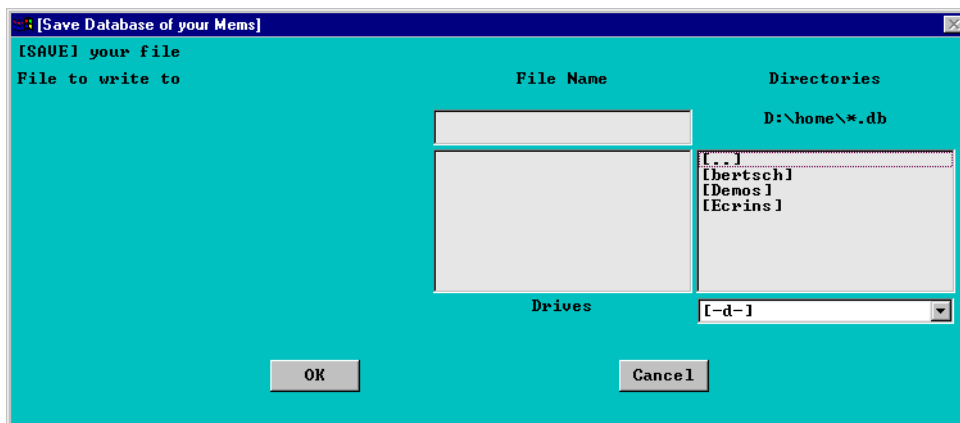


Figure 9: Save Database of your MEMS

- ☒ In the **Directories** list, choose the directory and in the **File Name** field, enter the name of the database (.db), then click **OK**.

## Unit

Before converting a database into a CIF file, define the **mcp\_unit** variable. If the 3D model loaded in the ANSYS session comes from the 3D Modeler, the unit is the micron.

- ☒ Enter *mcp\_unit=1.0e-6* in the **ANSYS Input** window (Figure 10).

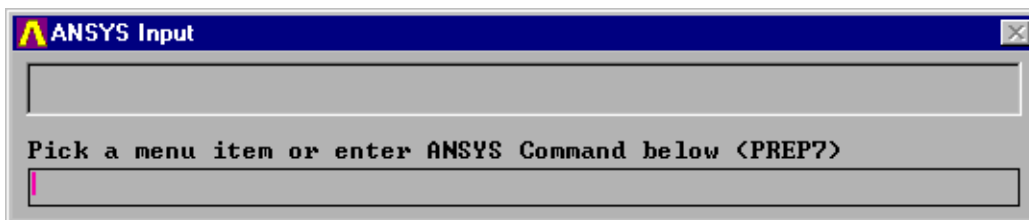


Figure 10: **ANSYS Input** window

## Exporting a CIF File

The **Export CIF file** option allows you to transfer a 3D solid model into a CIF file.

It creates a CIF file that any EDA tool can read. The 3D solid model must be loaded in the ANSYS active session as a database (.db), as an input file (.inp), or as a log file (.log). You can also create your volumes in the ANSYS graphics window.

To access this functionality, click the **LAYOUT** button in the ANSYS toolbar (refer to Section - The LAYOUT Menu Item).

For more details on this function, refer to Section - The Layout Generator Program.

## The LAYOUT Menu Item

In the ANSYS Toolbar, you can export a CIF file by clicking the **LAYOUT** button (see Figure 11).

If the **LAYOUT** button does not appear in the ANSYS toolbar, you can recover it by clicking **3-D to Layout** in the ANSYS Main Menu or by clicking **Clear & Start New** in the **File** menu of the ANSYS Utility Menu and choose the **Read file** option.

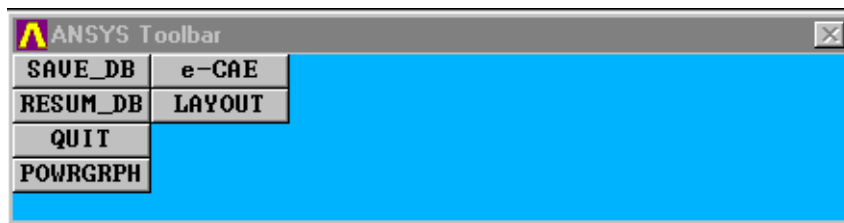


Figure 11: **LAYOUT** button in the ANSYS Toolbar

## The Layout Generator Program

Before exporting a CIF file, a 3D model must be represented in the active session of ANSYS.

In the layout (CIF file), the program creates all the layers of the 3D solid model defined in the previously specified technology file. If you do not want to load all the layers, refer to Definition of a Technology File.

If you click **LAYOUT** in the ANSYS Toolbar or **Export CIF File** in the **3D to Layout** palette, the **ANSYS to Layout** dialog box (Figure 12) prompts you to enter the appropriate information.

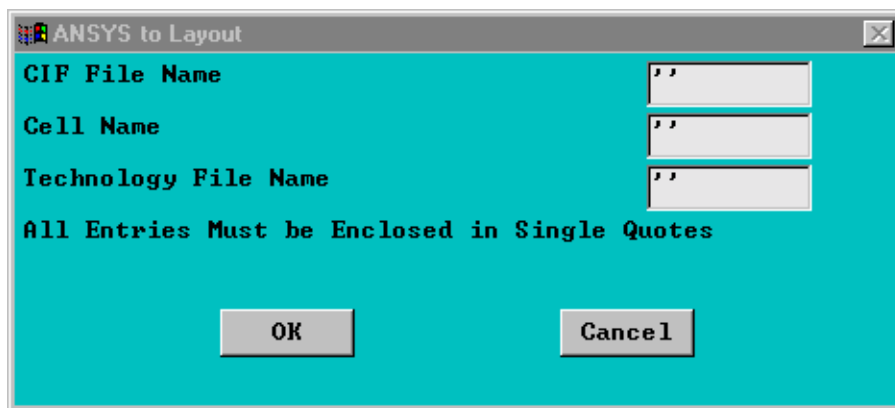


Figure 12: **ANSYS to Layout** dialog box

- ☒ In the **CIF File Name** field, enter the name of the CIF output file *without its extension*.
- ☒ In the **Cell Name** field, enter the name of the layout cell that will contain the layout.
- ☒ In the **Technology File Name** field, enter the name of the technology file.

**Note**

---

The technology file name should not be longer than 8 characters and should match the one you used when you started the session.

---

**Note**

---

All the entries must be between single quotes. These single quotes are already loaded in the dialog box. The CIF name and the cell name cannot contain more than 8 characters. If you enter a name including more than 8 characters, ANSYS only takes into account the first 8 characters. If the name contains a dot, the Layout Generator only takes into account the characters placed before the dot. For example, if the cell name is “demo.1”, the resulting cell name is “demo”.

---

The CIF file is created under the working directory you defined after launching ANSYS.

To import the CIF file in MEMS Pro, use the **Import Mask Data** options of the **File** menu in L-Edit.



Select **File > Import Mask Data**.

The **Import Mask Data** dialog box appears (Figure 13).

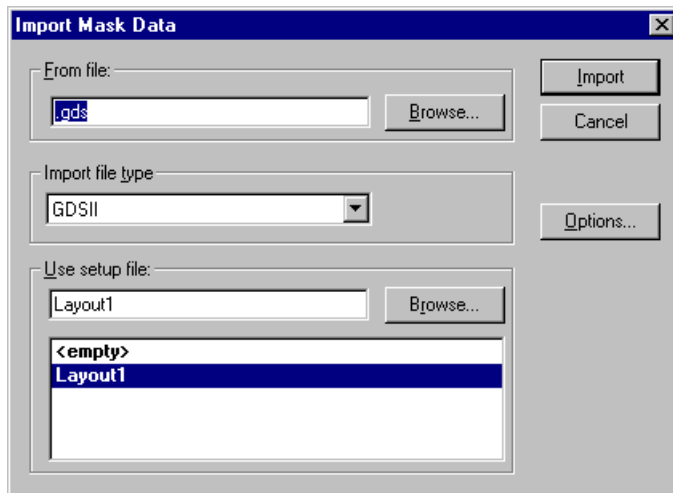


Figure 13: **Import Mask Data** dialog box

- ☒ You have the possibility to import a CIF or GDSII file. Make sure you chose CIF. By clicking **Import**, you open the layout view of the 3D model.

## Definition of a Technology File

This section provides information on how to create and use a technology file.

First of all, a technology file is used to attribute a CIF code to a volume which has a component name in ANSYS. A technology file is also used to detect negative masks and substrate, to get the layer name for the layout back and to define a component color in ANSYS.

To create a technology file, you need to know the layer name in the layout and if these layers are a negative mask or not. Then you have to define a component name, a CIF code and a color for each layer. Some layers do not appear in the layout but are important in the 3D-Model in ANSYS like OXIDE.

To create a technology file, first steps are:

- [1] Defining the layer name in the layout
- [2] Determining if the mask is negative or not
- [3] Choosing a CIF code for this particular mask.



The general syntax of a line in a technology file is the following:

```
Type   Name   "CIF code"   "CIF hole code"   COLOR   "Color Code"  
LAYOUT "Layout layer" END
```

The possibilities are the following:

- **Type**

SU for substrate,

NEG for a negative mask,

\* for a special layer,

A blank space or a tabulation should be used if it is a structural layer (associated to a positive mask).

- **Name**

Name of the component in ANSYS (max 8 characters).

- **CIF code**

A 3-letter abbreviation for the CIF file.

- **CIF hole code**

CIF code for the layer's holes.

- **COLOR**

Keyword / string for Color declaration in ANSYS (component rendering).

- **Color Code**

Color code for the layer. For holes, use a blank space.

- **LAYOUT**

Keyword for a Layout Layer. Specifies if the layer in 3D should be associated to a mask layer.

- **Layout layer**

The name of the layer in the layout.

- **END**

End of the line.

- Substrate

Within ANSYS, an ANSYS component called SUBSTRAT should exist.

It helps you detect what is not covered by the negative mask.

In the technology file, place the "SU" string at the beginning of the line describing this layer (we recommend you to start each technology file by this layer).

```
SU SUBSTRAT  COLOR CYAN  END
```

- Positive Mask

"Normal", (positive), layers do not have specific declaration.

If these layers contain holes, you must define a "hole layer" (associated to this positive layer) in the technology file. This layer should also have a layer name in the layout setup.

There is a specific declaration for this "hole layer". The "\*" character should be located at the beginning of the line describing this layer.

The following example shows you how the couple "positive layer / hole layer" is declared:

The structural layer with ANSYS component name POLY1, and CIF code CPS is RED in ANSYS and its layout layer name is POL1. Holes in this layer are mapped in a layout layer named HOLE1 which CIF code is CHO. Holes do not appear in the 3D view.

```
POLY1      CPS CHO COLOR RED  LAYOUT POL1      END
* HOLE1_WP CHO CPS                      LAYOUT HOLE1      END
```

If a mask is not in the layout, and if it has no holes, you must not write it in the technology file.

Considering a layer whose holes are not defined. If the program detects a hole, the layer name that is considered is the name of the layer containing the hole.

#### ▪ Negative Mask

This is a mask whose holes correspond to a layer in the layout.

Holes in this type of layer can be mapped to layers such as ANCHOR, VIA ...

Example: holes in an OXIDE layer can appear as CONTACT in the layout.

Place "NEG" to declare the negative mask before the ANSYS component name.

The following example represents an oxide component called OXI1 in ANSYS, with a CIF code AAA, and holes in the oxide mapped to the layer with CIF code COF. The ANSYS component is Yellow in ANSYS. This negative mask has no mask name in the layout:

```
NEG OXI1    AAA COF COLOR YELL  END
```

- Special Layers

Anchors, dimples, contact and via layers / components are commonly declared as special layers in the technology file. They are part of other structural layers such as Poly or Metal but at different heights.

In the technology file, enter the “\*” character at the beginning of the lines describing these layers.

- N diffusion, P diffusion

These are “normal” layers, except for those defined by combination of layers (example for act-area crossing poly to perform transistor).

The following is a technology file of the front side bulk etching process for ANSYS:

```

SU      SUBSTRAT                COLOR  CYAN                                END
*      ANCHOR    BAN    BPO    COLOR  RED      LAYOUT  anchor          END
      POLY1      BPO    BHO    COLOR  RED      LAYOUT  poly1          END
      METAL1     BME    BHM    COLOR  BLUE     LAYOUT  metal1          END
*      CONTACT   BCO    BME    COLOR  GREE     LAYOUT  contact          END
NEG     PASS      BPA    BPA    COLOR  YGRE     LAYOUT  pass            END
*      HOLPOLY1  BHO    BPO                                LAYOUT  holpoly1        END
*      HOLMETAL  BHM    BME                                LAYOUT  holmetal1       END
NEG     LOCOS     BLO    BAN    COLOR  WHIT                                END
NEG     OXIDE     BOX    BCO    COLOR  DGRA                                END

```

Hereafter is a technology file of the backetch process:

```

SU      SUBSTRAT                COLOR  CYAN                                END
      POLY1      PO1    HP1    COLOR  RED      LAYOUT  poly1          END
      POLY2      PO2    HP2    COLOR  MAGE     LAYOUT  poly2          END

```

	METAL1	ME1	HM1	COLOR	BLUE	LAYOUT	metal1	END
	METAL2	ME2	HM2	COLOR	CBLU	LAYOUT	metal2	END
	METAL3	ME3	HM3	COLOR	BMAG	LAYOUT	metal3	END
*	CONT_P12	CNT	PO2	COLOR	MAGE	LAYOUT	cont_p1p2	END
*	VIA1	VI1	ME1	COLOR	BLUE	LAYOUT	via1	END
*	VIA2	VI2	ME2	COLOR	CBLU	LAYOUT	via2	END
*	VIA3	VI3	ME3	COLOR	BMAG	LAYOUT	via3	END
NEG	OXIDE2	OX2	CNT	COLOR	YELL			END
NEG	OXIDE3	OX3	VI1	COLOR	ORAN			END
NEG	OXIDE4	OX4	VI2	COLOR	GREE			END
NEG	OXIDE5	OX5	VI3	COLOR	YELL			END
NEG	PASSIV	PAS	PAS	COLOR	YGRE	LAYOUT	passiv	END
NEG	LOCOS	LOC	LOC	COLOR	WHIT			END
	PWELL	PWE	PWE	COLOR	LFRA	LAYOUT	pwel	END
	ACTIVE-N	NPN	NPN	COLOR	DGRA	LAYOUT	nplus	END
	NPLUS	NPP	NPP	COLOR	DGRA	LAYOUT	nplus	END
	PPLUS	PPN	PPN	COLOR	MRED	LAYOUT	pplus	END
	ACTIVE-P	PPP	PPP	COLOR	MRED	LAYOUT	pplus	END

*	HOLE_P1	HP1	PO1	LAYOUT	holepoly1	END
*	HOLE_P2	HP2	PO2	LAYOUT	holepoly2	END
*	HOLE-M1	HM1	ME1	LAYOUT	holemetal1	END
*	HOLE_M2	HM2	ME2	LAYOUT	holemetal2	END
*	HOLE_M3	HM3	ME3	LAYOUT	holemetal3	END

The following is a technology file of the Surfmic process for ANSYS:

SU	SUBSTRAT			COLOR	CYAN			END
	POLY	POL	HPO	COLOR	RED	LAYOUT	poly	END
*	ANCHOR	ANC	POL	COLOR	RED	LAYOUT	anchor	END
*	DIMPLE	DMP	MET	COLOR	BLUE	LAYOUT	dimple	END
	METAL	MET	HME	COLOR	BLUE	LAYOUT	metal	END
*	CONTACT	CNT	MET	COLOR	GREE	LAYOUT	contact	END
*	HOLPOLY	HPO	POL			LAYOUT	holpoly	END
*	HOLMETAL	HME	MET			LAYOUT	holmetal	END



## Some requirements:

- 1) Place “SU” before the substrate, “NEG” before the negative mask or OXIDE, and “\*” before particular layers.

There should be a space or a tabulation before the other layers.

- 2) Name of the component in ANSYS should not be longer than 8 characters.
- 3) Except for the substrate that has no CIF code, the name of the CIF code should not be longer than 3 characters.
- 4) For the negative mask: indicate the CIF code of its hole.
- 5) For special layers: indicate the CIF code of the layer they belong to. In this example, for the ANCHOR layer, you have to indicate the CIF code of POLY1.
- 6) For the substrate: enter either a space or a tabulation.
- 7) The “COLOR” string

- 8) The ANSYS color code, except for hole layers and substrate, is the following:

MRED: Magenta-Red

CBLU: Cyan-Blue

YGRE: Yellow-Green

DGRA: Dark-Gray

MAGE: Magenta

CYAN: Cyan

YELL: Yellow

LGRA: Light Gray

BMAG: Blue-Magenta

GCYA: Green-Cyan

ORAN: Orange

WHIT: White

BLUE: Blue

GREE: Green

RED: Red

BLAC: Black

9) The “LAYOUT” string

If the layer does not appear in the layout, enter either a space or a tabulation.

10) Indicate the name of the layer in the layout, except if the layer does not appear in the layout.

11) Enter the “END” string to indicate the end of the line.

# Limitations

## Negative Mask Without Hole

For the time being, if a negative mask contains no holes, the resulting layout is not correct.

## Substrate

A substrate must appear in the 3D solid model of your active session.

## Splines

If the 3D is defined with splines instead of lines, the layout generator considers splines as straight lines.

## Boolean Operations on Layers

The tool does not handle boolean operations on layers.

# Tutorial

This tutorial illustrates how to use the ANSYS 3D-Model to Layout Generator tools.

For this tutorial, you will start from an existing model stored in an APDL file, import it in to ANSYS, modify it by adding volumes and deleting others, then generate the corresponding CIF file. The model you want to import is a micro mirror designed with the Surfmic technology.

Meshing and analyzing of the model is out of the scope of this tutorial.

## Import Memos

- ☒ Launch ANSYS 5.7 and select your working directory.

### **Note**

---

You can save all the necessary files under the same working directory. But, this is not mandatory.

---

- ☒ In the ANSYS Main menu, click **MEMSCAP Tools** (see Figure 14).

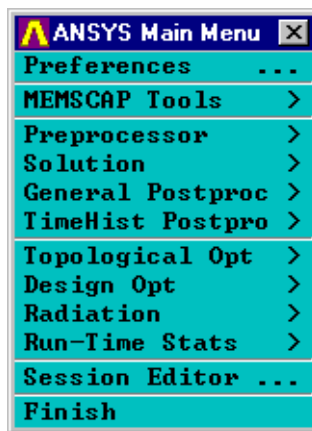


Figure 14: ANSYS Main menu

The **MEMSCAP Tools** menu appears:

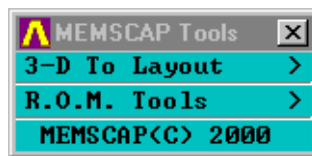


Figure 15: MEMSCAP Tools

- ☒ Then, click **3-D To Layout** (Figure 15).

The **Prompt** dialog box appears (Figure 16).

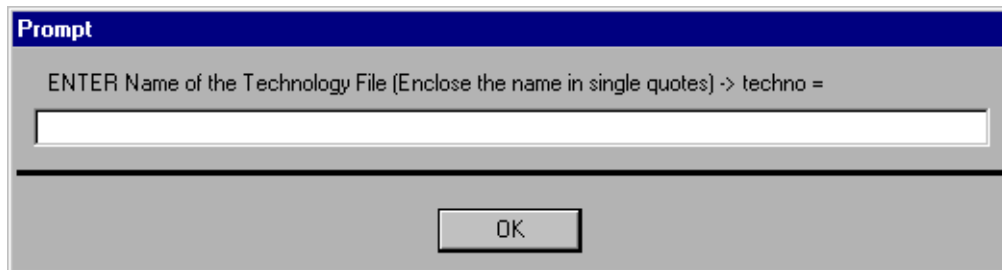


Figure 16: **Prompt** dialog

- ☒ Enter the technology file name, 'Surfmic'.

Remember that single quotes are necessary and that the name must not be longer than 8 characters.

- ☒ Click **OK**.

The **3-D To Layout** menu appears (Figure 17).

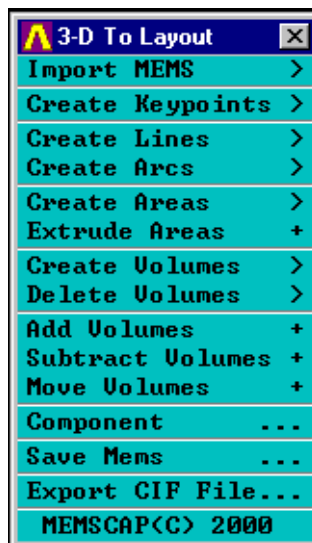
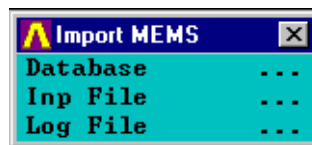


Figure 17: 3-D To Layout menu

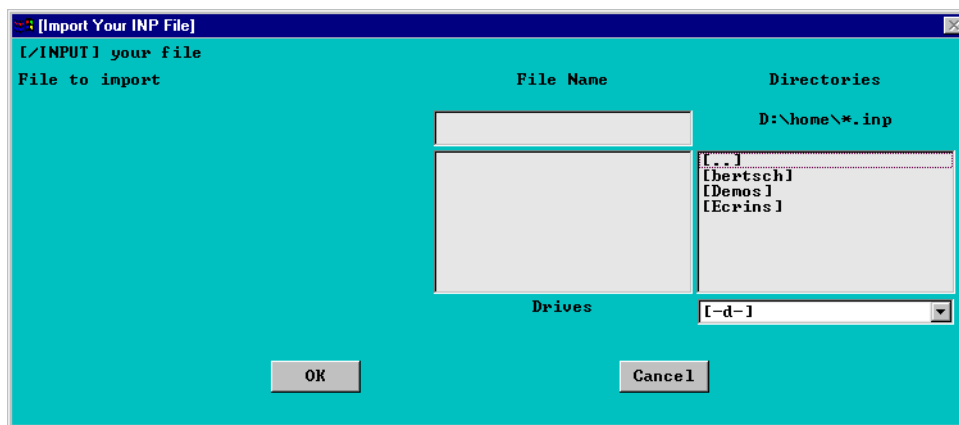
- ☒ Click **Import MEMS** to import a 3D model.

The **Import Mems** menu appears (Figure 18).



Figure 18: **Import MEMS** menu

- ☒ Click **Inp File**, and the following dialog box opens (Figure 19).

Figure 19: **Import Your INP File** dialog box

☒ Select the mirror.inp file under the appropriate directory.

☒ Click **OK**.

ANSYS warnings appear.

☒ Click **OK** to close the warning messages.

The 3D model mirror appears in ANSYS Graphics (Figure 20).

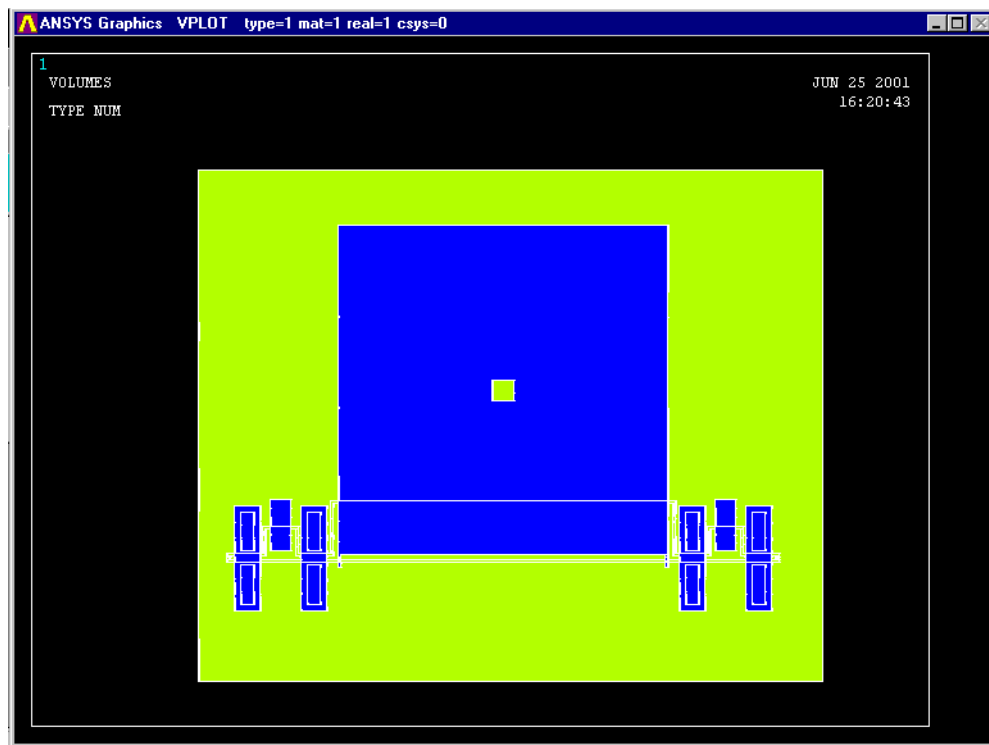


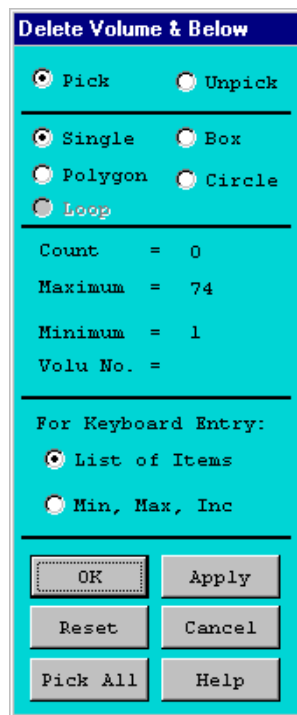
Figure 20: ANSYS Graphics window

## 3D Modifications

You will now delete parts of the mirror. These parts are blocks of the levers.

- ☒ Click **Delete Volumes > Volume & Below** in the **3-D To Layout** menu.

The **Delete Volume & Below** dialog box appears (Figure 21).

Figure 21: **Delete Volumes & Below** dialog box

- ☑ Click the **Pick** radio button.

The cursor has changed to an upward arrow.

- ☑ Select the volumes you want to delete (two blocks for each lever of the hinges) by clicking on them. The color of the selected part changes.

- ☑ Then, click **OK**.

The model appears in the ANSYS Graphics window (Figure 22).

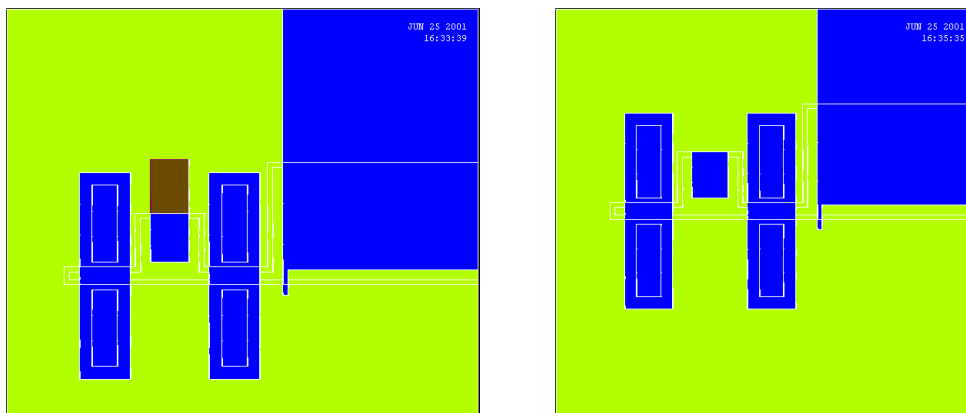


Figure 22: Mirror with and without levers

The hinges of the mirror now have smaller levers.

- ☒ Save the new mems file as a database (.db) using the **Save MEMS** menu item (see Figure 9), and then click **OK**.

## The Layout Generator Program

Before exporting a CIF file, define the `mcp_unit` variable in order to indicate the unit of the 3D Model.

- ☒ As the unit of the 3D model imported from MEMSCAP 3D Modeler is the micron, enter `mcp_unit=1e-6` in the **ANSYS Input** window (see Figure 23).

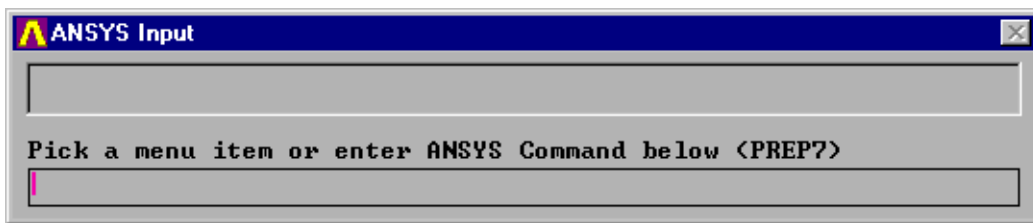


Figure 23: **ANSYS Input** window

In the layout (CIF file), the program creates all the layers defined in the corresponding technology file.

- ☑ Click **LAYOUT** in the ANSYS Toolbar (Figure 24), or **Export CIF Files** in the **3-D To Layout** menu.

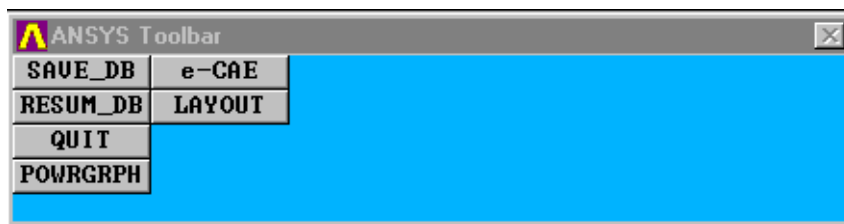


Figure 24: **ANSYS Toolbar** menu

The **ANSYS to Layout** dialog box opens (Figure 25). The first field refers to the name of the CIF file without extension, the second one to the name of the cell containing the layout, and the last field to the technology file name.



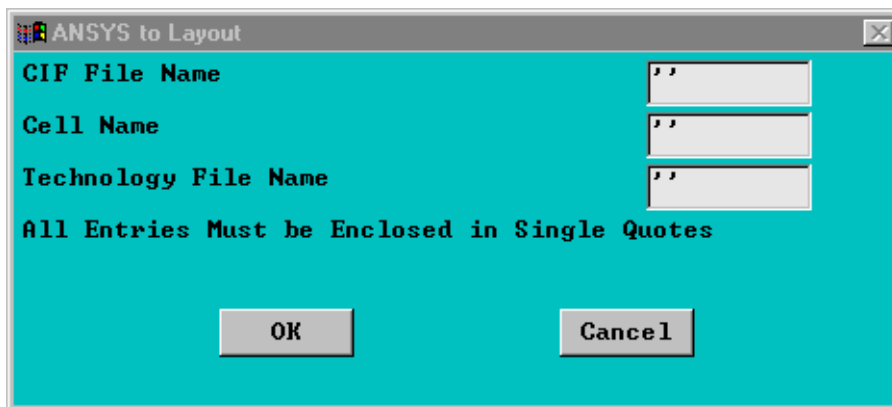


Figure 25: **ANSYS to Layout** dialog box

- ☒ Attribute a name to the CIF file (mirror), enter the name of the cell and the name of the technology file ('Surfmic')
  - ☒ Click **OK**.
- The mirror.cif file has been created in your working directory.
- You can now access this file in MEMS Pro.
- ☒ Launch L-Edit.

- ☑ Select **File > New** to create a new file.

The **New File** dialog box appears (Figure 26).

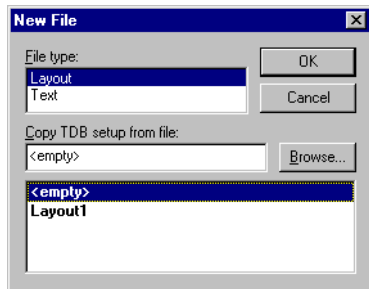


Figure 26: **New File** dialog box

- ☑ Choose *Layout* as **File Type** and *<empty>* in the **Copy TDB setup from file**.
- ☑ Click **OK**.
- ☑ Select **File > Replace Setup**.

The **Replace Setup Information** dialog box appears (Figure 27).

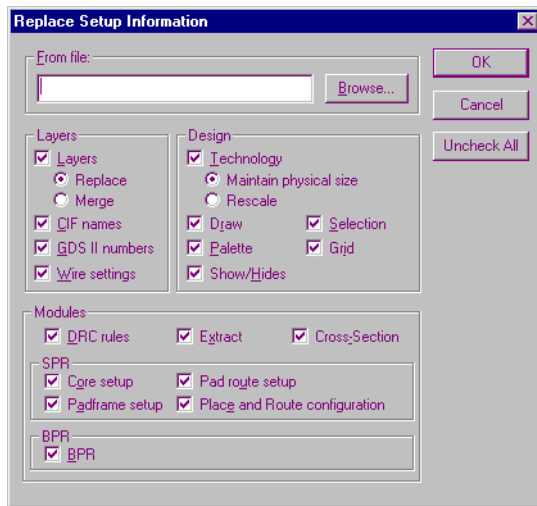


Figure 27: **Replace Setup Information** dialog box

- ☒ Browse for the ledit.tdb file that is located in the installation directory.
- ☒ Click **OK**.
- ☒ Select **File > Import Mask Data**.

The **Import Mask Data** dialog box appears (Figure 28).

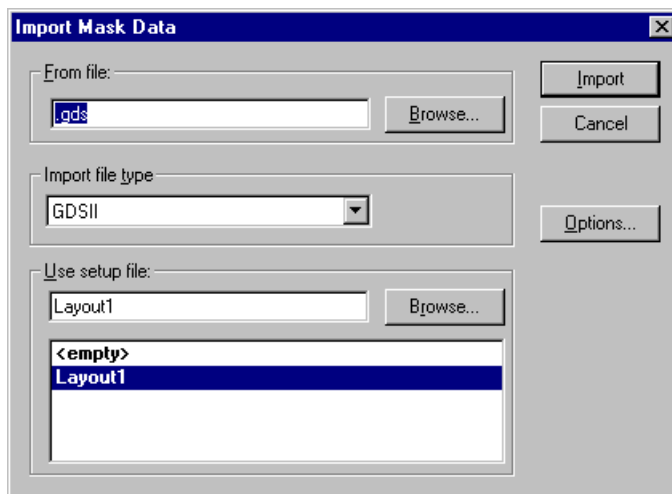


Figure 28: **Import Mask Data** dialog box

- ☒ Indicate the appropriate file type (CIF) in the **Import file type** field and browse for the mirror.cif file you have previously generated in ANSYS.
- ☒ Click **Import**.



The Layout of the mirror appears in the L-Edit window.



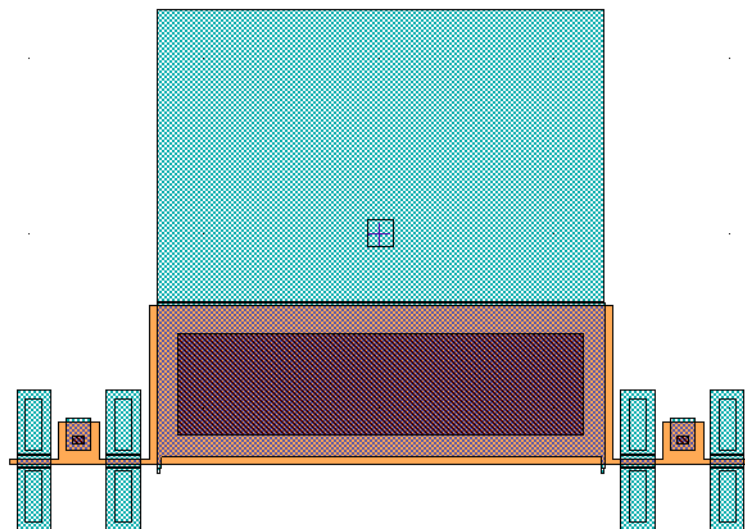


Figure 29: Layout view of the mirror

## 9 Reduced Order Modeling

---

- User Manual 335
- ROM Tutorial 370

# User Manual

## Introduction

The ROM (Reduced Order Modeling) tool is a MEMScAP customized ANSYS feature. It allows you to automatically generate behavioral models describing 3D structures reduced to a few master degrees of freedom. The resulting models describe the behavior of the considered degrees of freedom (e.g. displacement of nodes) according to variations of the applied loads (e.g. accelerations, voltages).

The behavioral models are written in SPICE and/or HDL-A<sup>TM</sup>\* languages. Additional formats will be implemented in further releases.

Finite element models may involve a large number of degrees of freedom so that full simulation, especially in the case of transient analyses, can be prohibitively expensive. As a consequence, designers really have to simplify models or limit the available results in order to obtain accurate but fast solutions. The solution is to create reduced order models from finite element models in a form that captures the essential physical behavior of a component and that is directly compatible with a system-level description.

\*HDLA is a trademark of Mentor Graphics Corporation



You can describe the dynamic behavior of a finite element model (assumed to be linear) using the following matricial equation:

$$[M]\{\ddot{x}\} + [C]\{\dot{x}\} + [K]\{x\} = \{f\} \quad (1)$$

The variables contained in the  $\{x\}$  arrays are called *degrees of freedom*. They entirely describe the state of the system. The number of DOFs can be very high ( $10^4$  to  $10^6$ ).

The  $[K]$ ,  $[M]$  and  $[C]$  matrices are respectively called the stiffness, mass and damping matrices and characterize the elastic behavior of the system, its inertia and damping effects.

The  $\{f\}$  array contains the equivalent forces related to structural variables.

One reduced order modeling approach, called reduction or condensation, consists in describing the behavior of the model by the following reduced set of equations:

$$[\hat{M}]\{\ddot{x}_R\} + [\hat{C}]\{\dot{x}_R\} + [\hat{K}]\{x_R\} = \{\hat{f}\} \quad (2)$$

## R.O.M. Menu

When running the MEMScAP customized ANSYS software, the following windows appear. They are the typical ANSYS windows. You can find help or information on them in the ANSYS help or in the ANSYS User Manual.

In the MEMScAP customized release of ANSYS, the **ANSYS Main** menu has been updated to give access to MEMScAP additional features via the **MEMSCAP Tools** button.

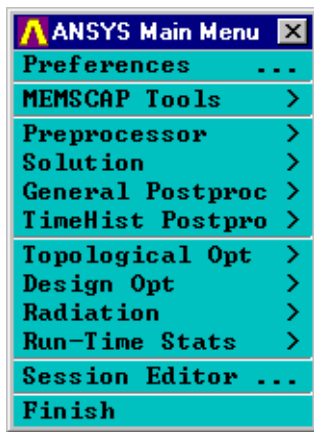


Figure 30: **ANSYS Main** menu

- ☑ To access the **MEMSCAP Tools** menu, click the **MEMSCAP Tools** option of the **ANSYS Main** menu.

A new window containing the **MEMSCAP Tools** menu opens.

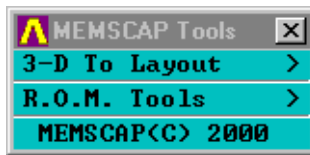


Figure 31: **MEMSCAP Tools** menu

- ☑ Click **R.O.M Tools** to access the Reduced Order Modeling main menu.

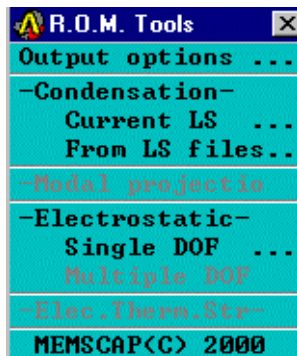
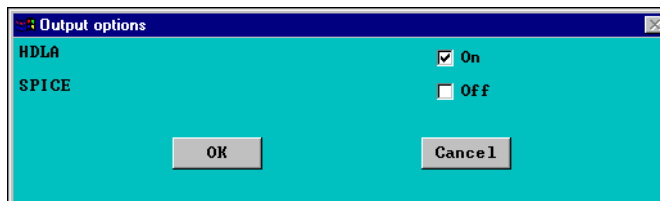


Figure 32: **R.O.M. Tools** menu

This menu gives you access to the reduction algorithms available in the MEMScAP tool. Algorithms that are greyed in the above window are not available at this time.

The first button, **Output options**, gives you access to a dialog box (Figure 33) allowing you to select the format(s) under which the reduced models will be generated.

Figure 33: **Output options** dialog box

At this time, two formats are available: HDLA™ and SPICE. The options can be independently checked or unchecked.

## Condensation Algorithm

The **Condensation** part of the **R.O.M. Tools** menu gives access to a classical reduction algorithm. It can deal with a single field as well as strongly coupled equations. For example, structural linear systems can be reduced according to this condensation algorithm.

### *Fundamentals*

The principle of this condensation method, usually known as the *static* or *Guyan* method, consists in selecting a reduced set of degrees of freedom that are assumed to be representative of the complete model behavior (*selected* degrees of freedom) and in eliminating the remaining degrees of freedom (*condensed* degrees of freedom) from the initial set of equations.

This condensation algorithm introduces the approximation that consists in assuming that the set of condensed degrees of freedom are related to the selected ones by the means of the static behavior equation.

In practice, the reduced model can only be connected to the external world using the set of selected degrees of freedom. In the same way, explicit forces can be applied to them, eventually combined with a linear combination of equivalent loads computed in accordance with the condensation algorithm.

This reduction method leads to an overestimation of the system eigen frequencies.

## *Running the Condensation*

### [1] Defining a model

Before using the MEMScAP R.O.M. tools, you have to load the finite element model of a structure in ANSYS. That means at least nodes, cells (possibly generated upon geometrical entities), requested parameters (if any) and physical properties as well as boundary conditions that are not included in the load case(s). There exist multiple methods to load a finite element model: executing a macro, introducing commands in the **ANSYS Input** window, using the GUI or combining the three previous methods.

### [2] Introducing loads

Regarding the theoretical basis of the algorithm, multiple load cases can be independently taken into account during the condensation process. Any given load case obtained by linear combination of the initial load cases can then be introduced in the reduced equations by combination of the reduced load arrays.

### [3] Selecting master degrees of freedom

You have to define one or more degrees of freedom to which the model will be reduced. In practice, these degrees of freedom are often those you want to concentrate on. Nevertheless, running the condensation algorithm introduces an approximation to the model behavior that is related to the choice of the master

degrees of freedom. You have to make sure that the selected master degrees of freedom are representative of the structural behavior of interest.

You can select the degrees of freedom (called *master* degrees of freedom in ANSYS) using the ANSYS **M** command that can also be accessed through the GUI (refer to the ANSYS documentation, for more information).

#### [4] Performing reduction

To easily manage the load cases, two buttons are available in the **R.O.M Tools** menu under the **Condensation** option: **Current LS** and **From LS file**. The **Current LS** option allows you to perform a condensation reduction with only one

load case and one degree of freedom. The **From LS file** allows you to perform a condensation reduction with multiple degrees of freedom and load cases.

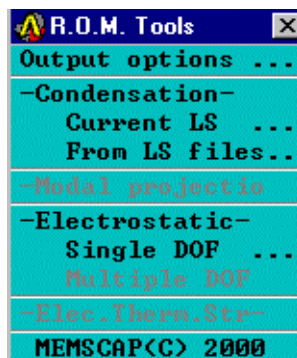


Figure 34: R.O.M. Tools menu



- **Current LS**

- ☒ If you click **Current LS**, the algorithm applies the currently defined load case, if any. In this case, a dialog box opens and prompts you to enter the output file name.

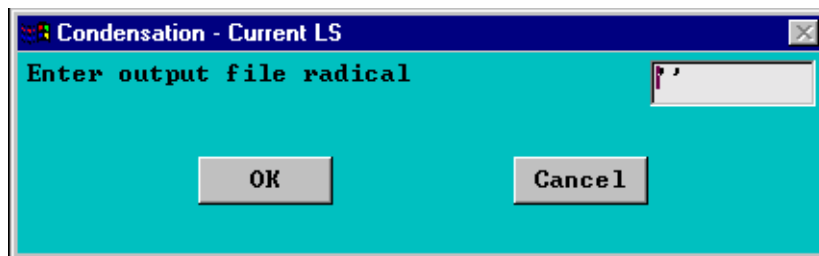


Figure 35: **Condensation - Current LS** dialog box

- ☒ Click **OK** to run the algorithm or select **Cancel** to close the dialog box.

**Note**

---

For more information on *performing the condensation reduction with one single DOF*, refer to Condensation: Reduction with Single DOF & Load Cases in this chapter.

---

- **From LS files**

- ☑ If you click **From LS files**, the algorithm applies multiple load cases. These load cases have to be previously defined in LS files (refer to the ANSYS documentation, for further information). The name of these files must be the current ANSYS jobname and the extension must contain the **s** letter followed by the LS file number.

In this case, a dialog box opens, prompting you to enter the output file name and a selection of LS files to process. This is done by defining the arguments of a loop.

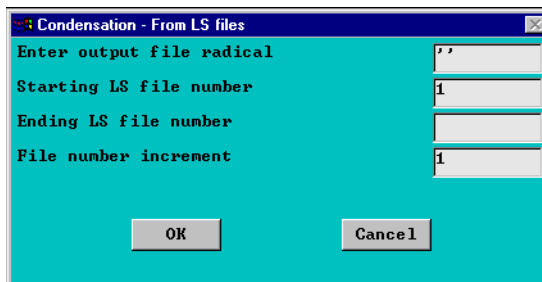


Figure 36: **Condensation - From LS files** dialog box

- ☑ Click **OK** to run the algorithm or select **Cancel** to close the dialog box.

**Note**

---

For more information on *performing the condensation reduction with multiple DOFs*, refer to Condensation: Reduction with Multiple DOFs & Load Cases in this chapter.

---

[5]      Generated files

In both cases, when running the algorithm, the ANSYS solver is called and as a consequence, classical ANSYS results and temporary files are created in your current working directory. During this process, the ANSYS jobname remains unchanged.

A temporary file named **scratch** is also generated.

Result files associated with the reduced model are created in your current working directory by adding the appropriate suffix to the output file prefix you previously entered. The following table provides the meaning of the different suffixes.

Suffix	Description
.sub	ANSYS substructure file
.hdla	Reduced model, described in HDLA™
.sp	Reduced model, described in SPICE

## Description of the .sp models

This type of (optionally) generated file contains a set of lumped basic circuit elements (resistors, capacitors, inductors, controlled voltage and current sources).

Depending on the number of load cases and degrees of freedom you entered, the number of nodes in the generated equivalent circuit can be high. Nevertheless, the number of nodes for simulation (external connecting nodes) remains low. These external connected nodes can be classified in three categories:

1. The input nodes: there are as many input nodes (called input1, input2...) as load cases. For SPICE simulations, the inputs used during ANSYS simulation must be applied to these entries as voltage excitation.

2. The output nodes: there are as many output nodes in an .sp files as degrees of freedom (e.g: with 3 degrees of freedom, you have 3 nodes called output1, output2 and output3). During system simulation, the voltage signals on these outputs are very close to the behavior of the corresponding degrees of freedom calculated by the FEM simulator.

3. The '0' node: this node must be connected to the voltage reference when the circuit model is instantiated in a larger SPICE circuit.

The following is an example of a generated one input (one load case), one output (one degree of freedom) .sp circuit:

\* T-Spice equivalent netlist generated by Memscap Model Builder

```
.param m11 = 1.151887e-06
.param d11 = 1.151887e-03
.param k11 = 4.116023e+01
.param l11 = 1.150145e-06
```

```
GM11 output1 0 output1_d2t 0 'm11'
GL11 0 output1 0 input1 'l11'
```

```
RK11 output1 0 '1/k11'
C_output1 N11 N12 1
E_output1 N11 0 output1 0 1
V_output1 N12 0 0
Vdamp_output1 output1 N13 0
FD11 N13 0 V_output1 'd11'
Hvel_output1 output1_dt 0 V_output1 1
L_output1 output1_d2t 0 1
Fvel_output1 0 output1_d2t V_output1 1
```

## Reduction of Electrostatically Coupled Structural Systems

The **Electrostatic** part of the **R.O.M. Tools** menu gives you access to an algorithm for the reduction of electrostatically coupled structural systems. In this algorithm, the system can only be reduced in terms of a single structural degree of freedom.

The reduced model consists of a scalar equation describing the transient behavior of the system, possibly combined with a relationship giving capacitance values in terms of the selected degree of freedom.

## *Fundamentals*

In the case of an electrostatic-structural coupled system, assumed to be linear and modeled by a finite element model, the equation governing its behavior can be written as follow:

$$\begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \ddot{x} \\ \dot{V} \end{Bmatrix} + \begin{bmatrix} C & 0 \\ 0 & 0 \end{bmatrix} \begin{Bmatrix} \dot{x} \\ V \end{Bmatrix} + \begin{bmatrix} K & 0 \\ 0 & K^V \end{bmatrix} \begin{Bmatrix} x \\ V \end{Bmatrix} = \begin{Bmatrix} f(x,V) \\ q \end{Bmatrix} \quad (3)$$

You can split the array of degrees of freedom into the {x} and {V} arrays which respectively describe the structural (displacements, rotations) and electrical (potential) state of the system.

The [K], [M] and [C] structural matrices are respectively called the stiffness, mass and damping matrices and characterize the elastic behavior of the system, its inertia and its damping effects.

The {f} array contains the equivalent forces related to the structural variables. The electrical equation is time-independent and expresses the irrotationality of

the electric field combined with Maxwell's divergence equation. The {q} array introduces a possible free charge loading of the structure.

These two sets of equations are said to be weakly coupled. That means off-diagonal terms of the matrices are null and coupling between the two physics arises from the load expressions. This kind of equations can only be solved using an iterative process.

Once the electric potential field equation is solved, the electric field can be deduced from the scalar potential and, on the external areas of conductors, the expression of electrostatic pressure can be evaluated. Structural equivalent forces are then computed by integration of the electrostatic pressure according to the structural degrees of freedom.

## [1] Structural behavior

Considering a structure that globally interacts electrostatically with its environment, the output of interest is a single degree of freedom.

From the structural point of view, the model behavior can be reduced to the following single scalar equation:

$$m\ddot{x}_R + c\dot{x}_R + kx_R = f(x_R, V) \quad (4)$$

The mass (m), the damping coefficient (c) and the stiffness (k) are computed by the application of the Guyan reduction algorithm, described in the previous section. This reduced model corresponds to a mass connected to a spring-damper.

The displacement dependent electrostatic interaction is included into the term of external force in which V represents the bias voltage applied between the structure and the environment.

## [2] Tuning eigen frequency

As the Guyan reduction algorithm leads to an overestimated approximation of the system eigen frequency, the reduced system parameters must be adjusted.

In the case of single degree of freedom systems, the eigen frequency is given in terms of the stiffness (k) and reduced mass (m) by the following expression:

$$\nu = \frac{1}{2\pi} \sqrt{\frac{k}{m}} \quad (5)$$

Computing the eigen frequency of the complete model by running a modal analysis and considering the reduced stiffness, the previous equation allows you to define a corrected reduced mass to obtain a reduced model that has an eigen frequency that matches exactly the eigen frequency of interest of the full model.

## [3] Electrostatic loads



The idea of this approach is to generate a numerical approximation of the external force term.

In order to obtain an expression of the last term of equation (4), representing the force versus displacement dependency, a numerical fitting is performed.

The procedure consists in generating a set of deformed configurations of the complete model. For each of them, the electrostatic loads acting on the structure are computed and condensed in accordance with the structural reduction algorithm.

In order to obtain an analytical expression of the previous result, a polynomial fitting is performed.

#### [4] Capacitance evaluation

The previous reduction method allows you to compute the structural system behavior taking into account the electrostatic coupling. Nevertheless, the output value of interest may be an electrical result, depending on the structural configuration.

To match this requirement, the generation of a capacitance expression in terms of the selected degree of freedom was added to the structural reduction procedure.

This generation is based on the same fitting procedure as the loads.

## *Running the Reduction Algorithm*

### [1] Defining a model

Before using the MEMScAP R.O.M. tools, you have to load a finite element model of your structure in ANSYS and provide all the data related to both electrical and structural behaviors of your system. That means at least nodes, requested parameters (if any) and physical properties as well as boundary conditions.

To provide a complete finite element model in ANSYS, generate a common geometrical model and separately describe the two independent physical environments (electrical and structural) using PHYSICS files (refer to the ANSYS User Manual and documentation, for further details).

The generation of the requested files and data can be performed using multiple methods: executing a macro, introducing commands in the **ANSYS Input** window, using the GUI or combining those three methods.

In the case of electrostatically coupled systems, the principle of solution is to iteratively solve the electrical or structural equation updating each equation using the results of the previous one. That means updating the geometry before solving the electrical equation and introducing electrostatic loads in the structural equation.

In practice, structural components are separated by gaps most often filled with air, that is a material that does not contribute to the structural stiffness calculation. So, it is not mandatory to take into account gap elements when solving the structural equation if it is possible to extrapolate displacement fields into the gaps in order to update the mesh of these zones. Such features are available in ANSYS but are not managed by the reduction algorithm.

As a consequence, to use the reduction tool, you must model gaps as structural components (possibly associated with negligible structural material properties). The structural fields are thus solved in the entire space. And the mesh update in gap zones occurs naturally with a greater efficiency than the one obtained using ANSYS mesh management features.

In the case of an electrical model featuring an open infinite boundary, it is convenient to model the far field behavior using a Trefftz domain (refer to the ANSYS documentation, for more information). When using such a feature, you have to carefully generate the model and PHYSICS files. Indeed, in the structural environment, the superelement associated with the Trefftz domain must be canceled (element type set to 0) and the associated constraint equations deleted, as well as infinite flags that become meaningless.

Furthermore, as the Trefftz domain includes a substructure that has a file name that is based on the current ANSYS jobname, it is strictly forbidden to use this jobname as an output file name.

## [2] Selecting Master Degrees of freedom

You have to define a single degree of freedom to which the model will be reduced. In practice, it is often the degree of freedom you want to concentrate on. Nevertheless, running the condensation algorithm introduces an approximation of the model behavior that is related to the choice of the master degrees of freedom. You have to make sure that the selected master degrees of freedom are representative of the structural behavior of interest.

You can select the master degree of freedom using the ANSYS M command that can also be accessed through the GUI (refer to the ANSYS documentation, for more information).

[3] Performing reduction

- ☑ To start the reduction algorithm, click the **Single DOF** button of the **Electrostatic** option.

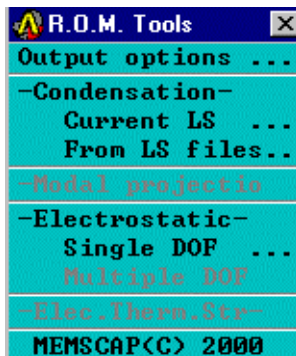


Figure 37: R.O.M. Tools menu

The following dialog box (Figure 38) opens, asking you to define data as well as algorithm control parameters. For some parameters, default values are suggested, but you must check their compatibility with your own configuration.

**Note**

Character strings must be in single quotes and must not be longer than 8 characters.

**Electrostatic-structure reduction to single DOF**

Enter output file radical

- structure PHYSICS -

Title

file radical 'structu'

file extension 'phy'

- electrostatic PHYSICS -

Title

file radical 'electric'

file extension 'phy'

- Electrical parameters -

Compute capacitance ☒ Yes

Conductors component name 'cond'

Number of conductors 3

Ground key Last conductor

Excitation component name 'cond2'

- algorithm parameters -

MINimal fitting value

MAXimal fitting value

Number of fitting points

Degree of fitting 4

Reference bias voltage 1

MAXimum number of eigen modes 0

OK Cancel

Figure 38: **Electrostatic - structure reduction to single DOF** dialog box

The **output file radical** is a prefix used to create result files (see generated files).

ANSYS Physics files are defined by a **title**, a **file radical** (prefix) and a **file extension**.

In accordance with your own model generation procedure, you have to define these three character strings for both the structural and electrical environments.

The reduction algorithm generates a scalar equation governing the transient response of the system in terms of the master degree of freedom. If required, a capacitance versus master degree of freedom relationship can also be generated. The capacitance values you extract are the lumped capacitances that are presented as matrix results. The size of the matrix depends on the number of conductors and the ground definition.

The generated capacitance relationship is time-independent. Transient capacitance values can be obtained by solving the reduced coupled structural equation and applying the capacitance relationship to the resulting master degrees of freedom values.

To activate or disable this relationship generation, check the **Compute capacitance** box.

From the electrical point of view, the system can be divided into conductors that electrostatically interact. To allow the algorithm to access them, a component of nodes must be associated to each conductor. The name of the component is the

**Conductors component name** defined in the dialog box followed by the **Number of conductors**. Component names must not be longer than 8 characters.

You must specify the **Number of conductors**. It is restricted to 2.

In an electrical system, a ground (bias voltage reference) must be defined. It may be associated with a modeled conductor or may be assumed to be at infinity. In the first case, the ground is assumed to be the last defined conductor (greatest conductor number).

Choose **Last conductor** or **At infinity** in the **Ground Key** field to specify the appropriate configuration.

To perform the reduction of the coupled system, the algorithm needs to apply electrical excitation to the system and evaluate the resulting electrostatic forces. This excitation is applied using the component of nodes specified in the dialog box as the **Excitation component name**. This component may be a conductor component.

It has been explained in the theoretical presentation of the algorithm that force and capacitance relationships are numerically computed point by point. An analytical expression is then extracted from these numerical values according to a mean square fitting method. This analytical expression is a polynomial in terms of the reduced degree of freedom, or its inverse, with a maximum degree (**Degree of fitting**) you can define in the dialog box.



The principle of the coupled system reduction algorithm is to analyze its electrical behavior in a set of structural configurations that covers the range of use. You must define this interval in terms of the master degree of freedom by giving its **Minimal** and **Maximal fitting values**. The number of configurations used for the sweeping range must also be specified (**Number of fitting points**).

### Note

---

The **Degree of fitting** must always be smaller than the **Number of fitting points**.

---

The **Reference bias voltage** is a control parameter that theoretically has no effect on the results but has been introduced to deal with possible numerical troubles. This value is used as the bias voltage applied to the excitation component to perform the coupled system analysis.

The **Maximum number of eigen modes** is an advanced parameter related to the reduced model frequency response tuning process.

The theoretical presentation of the Guyan condensation algorithm explained that this method introduces an approximation on the mass matrix that leads to an overestimated approximation of the eigen frequencies. To overcome this behavior, a modal analysis of the structural system has been performed and a corrected reduced mass value has been computed to match a given eigen frequency.

As an upper limit of the eigen frequencies is of interest (initial reduced eigen frequency), the modal analysis is performed by imposing a maximum number of

modes to compute (the default is 10) as well as the upper limit of the frequencies. This works correctly in the average case, but the procedure could nevertheless fail in some particular cases.

Regarding the procedure, it is obvious that the maximum number of eigen modes to compute must be greater or equal to the structural eigen mode of interest (related to the master degree of freedom).

A second case of failure is a system in which the frequency gap between the mode of interest and the following one is less than the error on the eigen frequency induced by the condensation algorithm. In this case, you must indicate the number of the mode to be considered in the tuning process. Nevertheless, in such a case, the reduced structural model will probably be unable to accurately represent the transient response of the real system.

- ☒ Once you have completed the previous dialog box, click **OK** to run the algorithm or select **Cancel** to close the dialog box.

#### [4] Algorithm output

When the reduction algorithm is running, information is printed in the **ANSYS Output** window. The display is related to the algorithm status as well as results and accuracy estimation.

Hereafter, is an example of display during the execution of the algorithm.

The first information displayed is the title of the algorithm and a summary of the data you entered in the previous dialog box.

```
*****
Electrostatic reduction (single degree of freedom)
*****

Structure PHYSICS:
File    = "structu.phy"
Title   = ""

Electrostatic PHYSICS:
File    = "electric.phy"
Title   = ""

Conductors name           = "cond"
Number of conductors      = 2
Ground key                 = 0
Capacitance matrix dimension = 1
Excitation component      = "cond2"
```

The next display is related to the structural condensation algorithm.

The name of the ANSYS substructure file is printed as well as the reduced values of mass, damping and stiffness.

```
=====
Run structural substructuring
```

```

=====

ANSYS substructure file = "tmp.sub"

Waiting for ANSYS solution ... Done

Structural reduced parameters:

Stiffness      (k) = 1.1646848e+00
Mass           (m) = 4.9866184e-12
Damping        (c) = 0.0000000e+00

```

Then, comes the modal analysis and the frequency response tuning procedure.

Structural eigen frequencies computed on the complete model appear, followed by a comparison between the estimated eigen frequencies and mass values.

The comparison of the values gives an indication on the representation of the mode by the selected master degree of freedom.

```

=====

Run structural Modal analysis

=====

Waiting for ANSYS solution ...Done

-----
Mode Frequency
-----

```

```

1  7.6670774e+04
-----

```

Tuning transient response:

```

Expected      eigen frequency = 7.6670774e+04
Approximated  eigen frequency = 7.6916812e+04 (0.3209 % shift)
Reduced mass   = 4.9866184e-12
Corrected mass = 5.0186740e-12 (0.6428 % shift)

```

The next output indicates the sweeping parameters for coupled effects evaluation.

During these analyses, a status is printed after each set of analyses.

```

=====
Perform coupled analyses
=====

Number of analyses      = 10
Degree of fitting       = 4
Minimum DOF value       = -2.0000000e-06
Maximum DOF value       = 1.0000000e-06
Reference bias voltage   = 1.0000000e+00

--> Performing set of coupled analyses number 1/10
--> Performing set of coupled analyses number 2/10
--> Performing set of coupled analyses number 3/10
--> Performing set of coupled analyses number 4/10

```

```

--> Performing set of coupled analyses number 5/10
--> Performing set of coupled analyses number 6/10
--> Performing set of coupled analyses number 7/10
--> Performing set of coupled analyses number 8/10
--> Performing set of coupled analyses number 9/10
--> Performing set of coupled analyses number 10/10

```

At the end of the coupled effect evaluations, a summary of analysis points is printed.

If the capacitance relationship is requested, the mutual capacitance value between first and second conductors (Capa[1,2]) is printed in the result table, or the capacitance between the conductor and the ground (Capa[1,1]) if a single conductor is modeled.

Point	Master DOF	Reduced FMAG	Capa[1,1]
1	-2.0000000e-06	-2.5765530e-09	1.2358000e-14
2	-1.6666667e-06	-1.8065974e-09	1.0862294e-14
3	-1.3333333e-06	-1.3646935e-09	9.7759145e-15
4	-1.0000000e-06	-1.0811644e-09	8.9370341e-15
5	-6.6666667e-07	-8.8548790e-10	8.2623515e-15
6	-3.3333333e-07	-7.4327955e-10	7.7036706e-15
7	4.2351647e-22	-6.3585629e-10	7.2307667e-15
8	3.3333333e-07	-5.5223359e-10	6.8235294e-15
9	6.6666667e-07	-4.8555243e-10	6.4679561e-15
10	1.0000000e-06	-4.3132108e-10	6.1539385e-15

Once the numerical values of reduced forces and capacitance (if requested) have been approximated by an analytical expression, an accuracy estimation of the fitting process is performed on each fitted value.

This evaluation consists in comparing, at each fitting point, the numerical value with the analytical expression.

A summary of the greatest absolute difference is also printed and compared with the absolute mean value of the numerical values in order to obtain a relative data.

#### [5] Generated files

In both cases, when running the algorithm, ANSYS solvers are called and as a consequence, files are created in your current working directory. During this step, the ANSYS jobname remains unchanged.

While result files are created, “scratch”, “cmatrix.out” and “jobname.s\_db” files are generated.

Result files associated to the reduced model are created in your current working directory by adding the appropriate suffix to the output file radical prefix you entered. The following table provides the meaning of the different suffixes.

Suffix	Description
.sub	ANSYS substructure file

.hdla	Reduced model, described in HDLA™
_capa.hdla	Capacitance relationship, described in HDLA™
.sp	Reduced model, described in SPICE
_capa.sp	Capacitance relationship, described in SPICE

[6] Behavioral model

## Description of the .sp files

When the SPICE output option is active, two .sp files can be generated.

- The first .sp file is created by default. The file name prefix is the name you entered or the ANSYS jobname if no file radical prefix is specified. It contains a circuit of lumped elementary elements used to model the Master Degree Of Freedom as a function of the applied voltage. The connection to a global circuit of this model has to be performed through three nodes: input1, output1 and 0. The signal connected to the input1 node must be the applied voltage. The voltage at the output1 node models the master degree of freedom behavior. The 0 node must be connected to the voltage reference in the circuit.
- The second .sp file is generated only if the capacitance output option is toggled to **Yes**. The file name prefix consists of two parts separated by an underscore. The first part is identical to the previous file prefix and the



second part is the “capa” string. The connection to an external global circuit is similar to the MDOF behavior connection. There is only one difference: instead of having only one output node in the MDOF model case (output1), you can have up to NM output nodes in the capacitance model (output1, ..., outputNM), NM being the number of mutual capacitance interaction coefficients. In this release, NM=1.

The following provides an example of a capacitance output:

```
* Spice equivalent netlist generated by Memscap MEMS Modeler
.param force_polynome0 = -1.538212e+10
.param force_polynome1 = -3.855341e+15
.param force_polynome2 = -1.216000e+20
.param force_polynome3 = 3.139320e+24
.param force_polynome4 = -2.171035e+29
.param m11 = 5.468008e-13
.param d11 = 0.000000e+00
.param k11 = 4.947165e-01
RK11 output_f1 0 '1/k11'
C_output_f1 N11 N12 1
E_output_f1 N11 0 output_f1 0 1
V_output_f1 N12 0 0
Vdamp_output_f1 output_f1 N13 0
FD11 N13 0 V_output_f1 'd11'
Hvel_output_f1 output_f1_dt 0 V_output_f1 1
L_output_f1 output_f1_d2t 0 1
Fvel_output_f1 0 output_f1_d2t V_output_f1 1
GM11 output_f1 0 output_f1_d2t 0 'm11'
```

```

Goutput_f1 0 output_f1 VALUE = {(v(input1)*v(input1))/((-
1.538212e+10)+(-
3.855341e+15)*v(output_f1)+(-
1.216000e+20)*v(output_f1)*v(output_f1)+(3.139320e+24)*v(output_f1)
*v(output_f1)*v(output_f1)+(-
2.171035e+29)*v(output_f1)*v(output_f1)*v(output_f1)*v(output_f1))}
Eoutput1 output1 0 VALUE = {1.0/
((5.647164e+14)+(4.949925e+19)*v(output_f1)+(-
1.915843e+24)*v(output_f1)*v(output_f1)+(2.133310e+29)*v(output_f1)
*v(output_f1)*v(output_f1)+(-
2.785264e+34)*v(output_f1)*v(output_f1)*v(output_f1)*v(output_f1))}

```

# ROM Tutorial

This tutorial aims at briefly explaining how to use the R.O.M. (Reduced Order Modeling) tool.

Finite element models may involve a large number of degrees of freedom so that full simulation, especially in case of transient analyses, can be prohibitively expensive. The aim of the R.O.M. tool is to make model simplifications or limit the available results in order to obtain accurate, but fast solution during simulations.

The following parts indicate how to perform a reduction with two different examples: condensation of an accelerometer model and reduction of an electrostatic-structural coupled system.

## Condensation: Reduction with Single DOF & Load Cases

Before using the R.O.M tool, you first have to provide a finite element model in ANSYS. For the following examples, use an inertial accelerometer model, that consists of a structural mass supported by four thin beams clamped at their extremities, as shown in Figure 39.

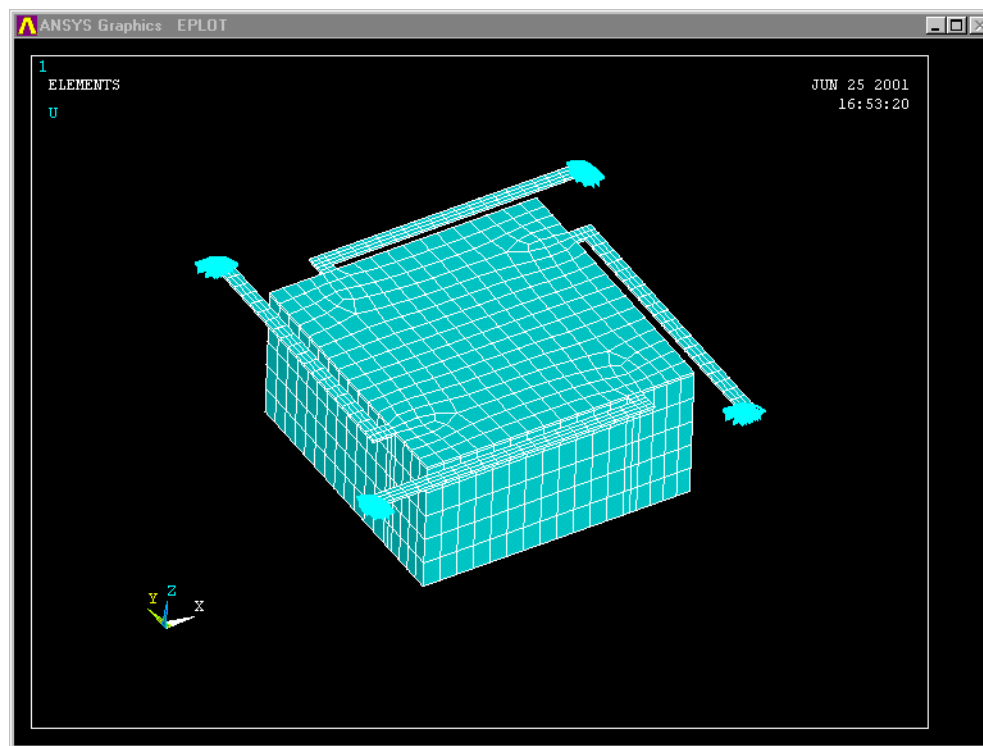


Figure 39: ANSYS Graphics window

The first example is the most simple. A single load case is taken into account and the accelerometer model is condensed in terms of a single degree of freedom.

## *Model Generation*

First, provide a finite element model of your structure. In that example, all the information is gathered in two macros; you only have to execute them. This can be done using multiple tools.

All the information concerning the model (geometry, mesh and boundary conditions) is gathered in the file called `accelman.mdl` (located under the **tutorial** directory).

- ☒ Copy `accelman.mdl` to your working directory.
- ☒ In the **ANSYS Input** window, enter the following command (see ANSYS documentation, for further details):

**\*USE,accelman.mdl** (Figure 40)

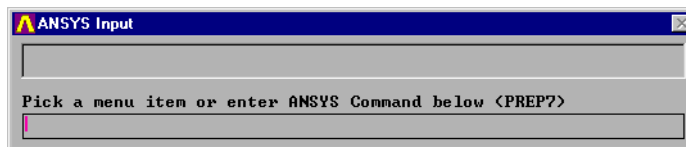


Figure 40: **ANSYS Input** window

The accelerometer appears in the ANSYS Graphics EPLOT window (Figure 39). It is a 3D meshed view of the accelerometer.

The load case is an acceleration applied to the vertical axis of the model.

The conditions corresponding to the chosen load case are contained in the file called accelman.load1 (located in the **tutorial** directory).

- ☒ Copy the following macro accelman.load1 to your working directory.
- ☒ Click the **ANSYS Input** window and execute the macro, using the ANSYS **\*USE** command:

```
*USE, accelman.load1
```

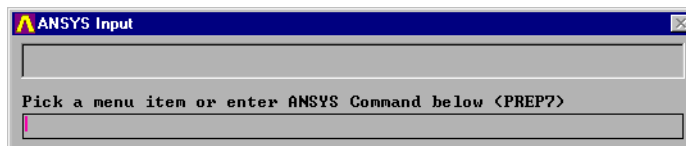


Figure 41: **ANSYS Input** window

The applied acceleration is shown by an arrow on the triad (Figure 42). It allows you to verify the applied boundary conditions.

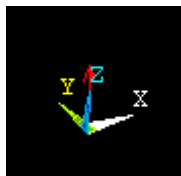


Figure 42: Arrow in the Z direction

### *Performing Reduction*

You now have to define one or more degrees of freedom to which the model will be reduced.

In this example, you are interested in the model behavior at a particular node that has a number given to the N\_MASTER variable (see the model description macro: accelman.mdl). In fact, it is the center node of the accelerometer top face. We will use only one degree of freedom: the vertical displacement.

- ☒ Define this degree of freedom as a master degree of freedom by entering the following command in the **ANSYS Input** window:

**M, N\_MASTER,UZ.**

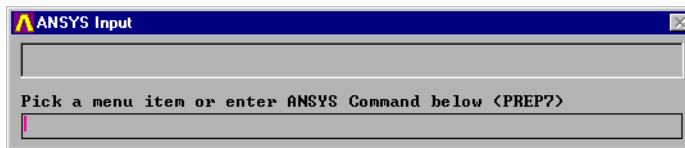


Figure 43: **ANSYS Input** window

A symbol appears on the selected node to indicate the selected degree of freedom.

- ☒ Select **MEMSCAP tools > R.O.M Tools** (Figure 44) in the **ANSYS Main** menu.



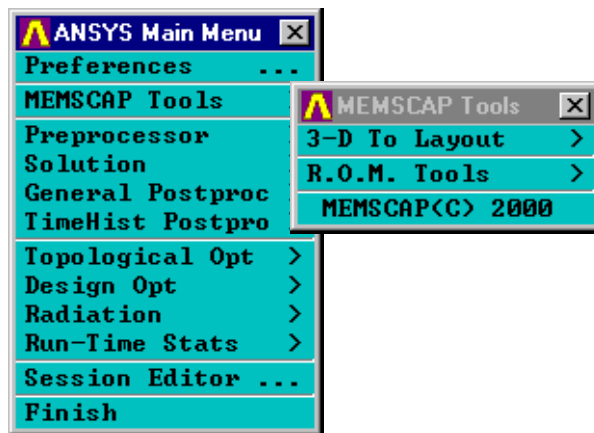


Figure 44: R.O.M. Tools menu

The **R.O.M. Tools** window appears and gives access to all the condensation algorithms implemented in the MEMSCAP R.O.M. tool.

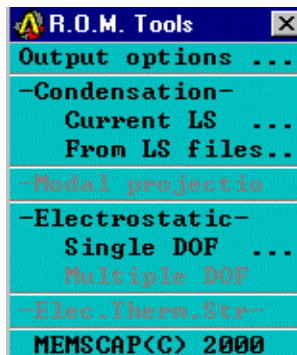


Figure 45: **R.O.M Tools** menu

Before performing the reduction, you must select the format(s) for which the reduced models will be generated.

- ☑ Click **Output options**.

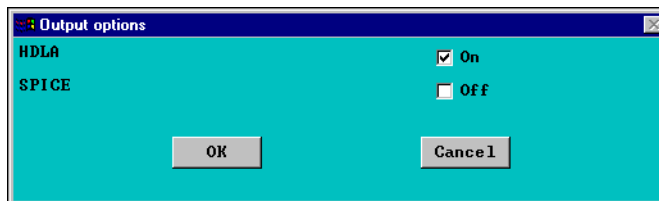


Figure 46: **Output options** dialog box

The **Output options** dialog box allows you to save your model in SPICE, HDLA™, or both languages.

As an example, the SPICE language is chosen.

- ☑ Click **SPICE**.

Now, you may run the condensation. The condensation part of the R.O.M. tool gives access to the Guyan-Irons reduction algorithm.

- ☑ In the **R.O.M Tools** menu, click **Condensation > Current LS**, to work on the current load case.

The **Condensation - Current LS** window appears (Figure 47).

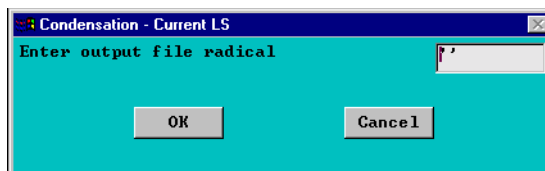


Figure 47: **Condensation - Current LS** dialog box

The algorithm applies the currently defined load case. The above dialog box asks you to enter the output file name. For example, call it 'MyExample1'.

### Warning

---

*Be careful to enclose the output file name in single quotes.*

---



Click **OK** to run the algorithm.

During the execution of the algorithm, the **ANSYS Process status** progress bar indicates to you which action the software is performing. Moreover, information is printed in the **ANSYS Output** window, informing you about the currently performed task and its results.

```
*****
Guyan-Irons condensation
*****
Ansys substructure file = "MyExample1.sub"
```

```

Run substructuring with current load step
-----
Waiting for ANSYS solution ... Done
Number of selected DOF = 1
Number of load steps = 1
*****
Reduced model successfully generated !
*****

```

After the execution, results files are created in your working directory. They contain the behavioral model (reduced equation) written in the selected format. In this case, the output file is MyExample1.sp.

Another file, called MyExample1.sub, has been created. It is the ANSYS substructure file.

## Condensation: Reduction with Multiple DOFs & Load Cases

The second example uses the same model (the accelerometer). It is more complex than the first example. The aim is to apply three load cases to the model, and select multiple degrees of freedom. The loads are accelerations along the three axes (X, Y and Z).

You will model the displacement of one node (located in the middle of the proof mass) in the X, Y and Z directions resulting from an acceleration in an unknown direction.

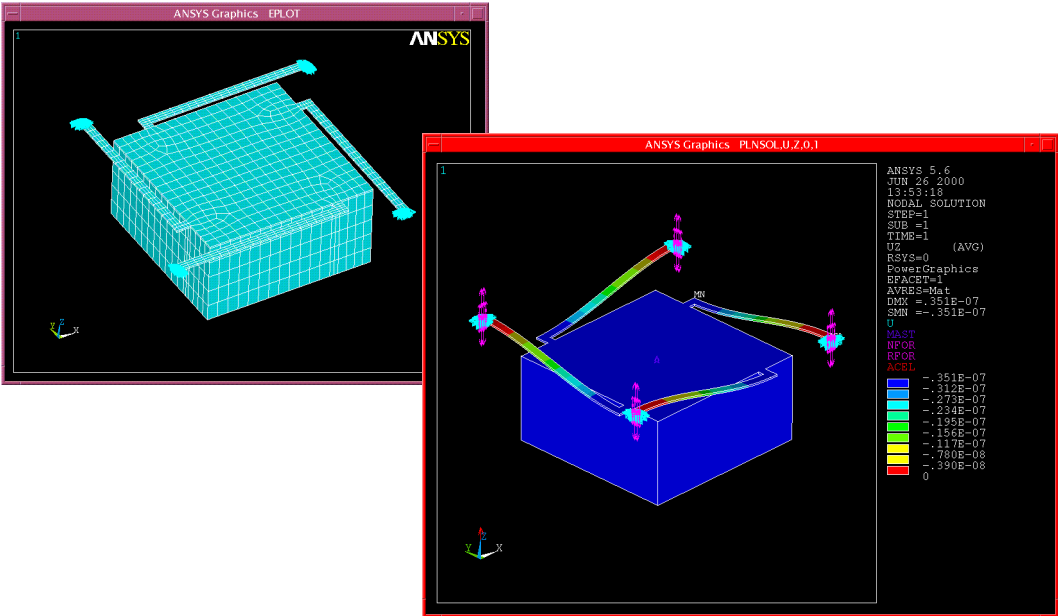


Figure 48: Model description

## Model Generation

First, provide a finite element model of your structure. In that example, all the information is gathered in two macros; you only have to execute them. This can be done using multiple tools.

All the information concerning the model (geometry, mesh and boundary conditions) is in the file called “accelman.mdl” located under the **tutorial** directory.

- ☒ Copy accelman.mdl under your working directory.
- ☒ Click the **ANSYS Input** window and enter the following command (see ANSYS documentation, for further details).

```
*USE, accelman.mdl
```

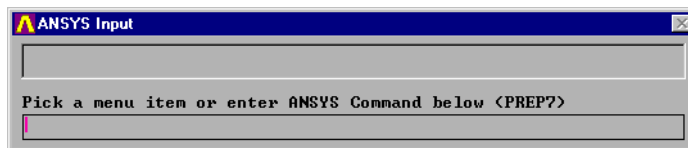


Figure 49: **ANSYS Input** window

The following window appears:

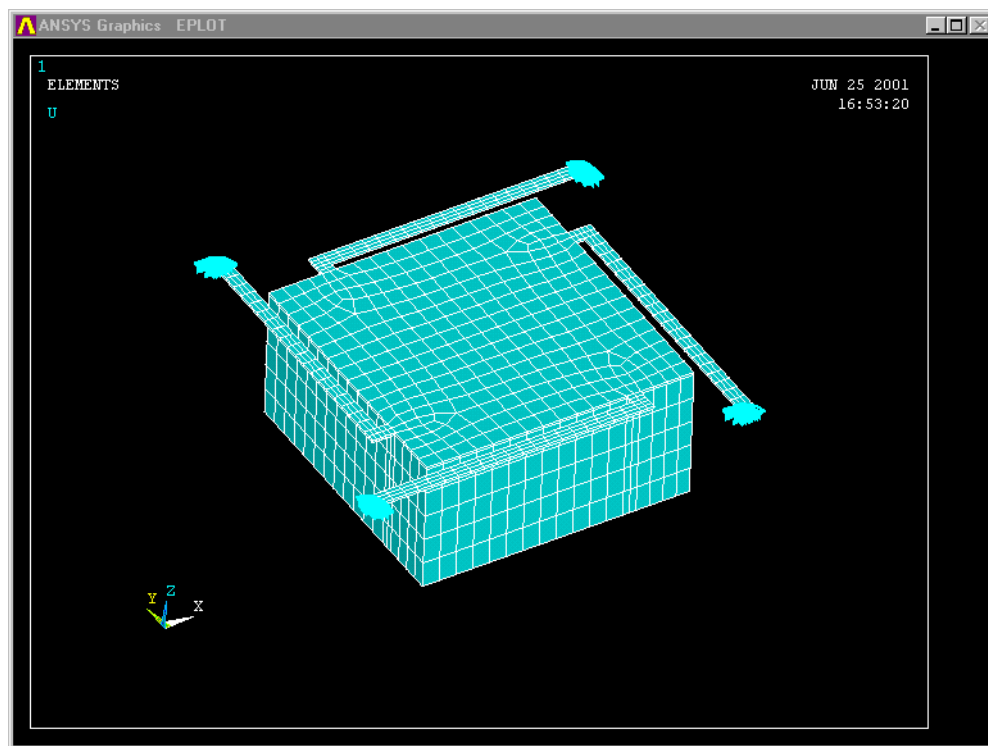


Figure 50: **ANSYS Graphics EPlot** window



The **ANSYS Graphics EPlot** window shows the 3D view of the accelerometer, and its mesh.

The conditions corresponding to the chosen load cases are contained in the macro called `accelman.load3` (located under the **tutorial** directory). It generates three LS files in the working directory.

- ☒ Copy `accelman.load3` under your working directory.
- ☒ Click on the **ANSYS Input** window, and execute the macro.

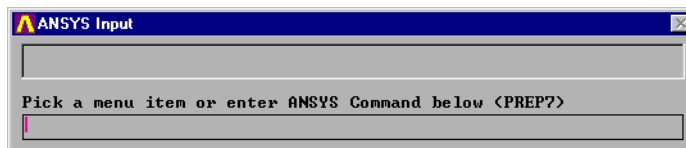


Figure 51: **ANSYS Input** window

The macro generates three load cases and, for each of them, loads the configuration to a file.

After execution, the last load case remains defined in ANSYS. In this case, an acceleration in the Z direction is symbolized by an arrow on the triad.

## Performing Reduction

You now have to define one or more degrees of freedom to which the model will be reduced.

In this example, you are interested in the model behavior at a particular node that has a number attributed to the `N_MASTER` variable (see the model description macro, `accelman.mdl`). In fact, it is the center node of the accelerometer top face. Choose three degrees of freedom, which are the displacements along the three axes.

- ☒ Define these degrees of freedom as master degrees of freedom by entering the following command in the **ANSYS Input** window.

**M,N\_MASTER,ALL**

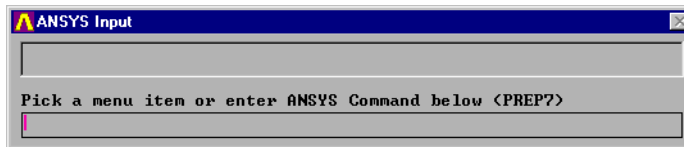


Figure 52: **ANSYS Input** window

As the degrees of freedom associated with the nodes are the displacements in the three directions of the space, you can select the three degrees of freedom using a single command.

ALL means that you now consider all the degrees of freedom associated to the N\_MASTER node.

A symbol is displayed on each selected degree of freedom.

- ☑ Click **MEMSCAP Tools > R.O.M Tools**.

The **R.O.M. Tools** window appears and gives access to all the condensation algorithms implemented in the MEMSCAP Tools.

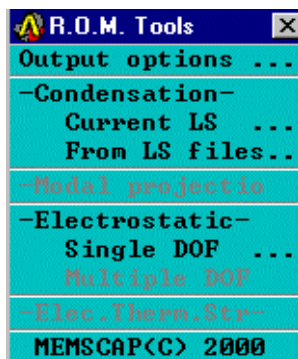


Figure 53: **R.O.M. Tools** menu

Before performing the reduction, you must select the format(s) in which the reduced models will be generated.

- ☑ Click **Output options**.

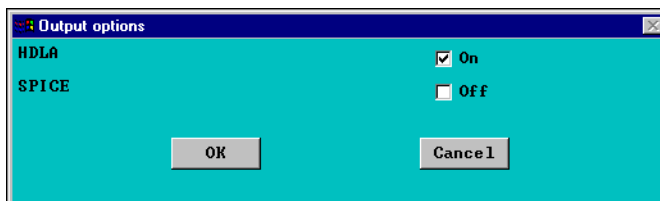


Figure 54: **Output options** dialog box

The **Output options** dialog box allows you to select HDLA™, SPICE or both languages for the results file.

In this example, the SPICE language is chosen.

- ☑ Click **SPICE**.

Now, you may run the condensation. The condensation part of the R.O.M. tool gives access to the Guyan-Irons reduction algorithm.

- ☑ As you are working with three load cases, click **Condensation > From LS files** in the **R.O.M Tools** menu.

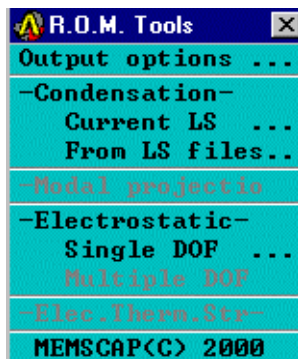


Figure 55: Selecting the **Condensation from LS files** option

The **Condensation - From LS files** dialog box appears.

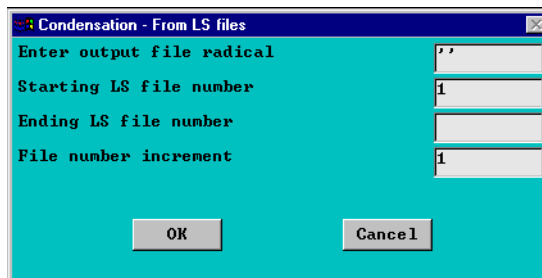


Figure 56: **Condensation - From LS files** dialog box

- ☒ Set the output file radical prefix to 'MyExample3'.

### Warning

---

*Do not forget to enclose the output file radical prefix in single quotes.*

---

- ☒ Set the **Ending LS file number** to 3.
- ☒ The algorithm applies three load cases. These load cases have been previously defined in LS files. The name of these files are the current ANSYS jobname, followed by the number of the LS file.

The created LS files are called <working\_directory\_name>.s01, <working\_directory\_name>.s02 and <working\_directory\_name>.s03.

- ☒ Click **OK** to run the algorithm.

During the execution, information is printed in the **ANSYS Output** window, informing you about the currently performed tasks and their results.

```
*****
Guyan-Irons condensation
*****
```

```
Ansys substructure file = "MyExample3.sub"
```

```
Run Substructuring with LS files
-----
```

```
LS files      FROM    1
              TO      3
              INC      1
```

Waiting for ANSYS solution ... Done

Number of selected DOF = 3

Number of load steps = 3

```
*****
```

Reduced model successfully generated !

```
*****
```

After execution, result files are created in your working directory. They contain the behavioral model (reduced equation), written in the selected format. In this case, the output file is MyExample3.sp.

The following figure gives an explanation of the model.

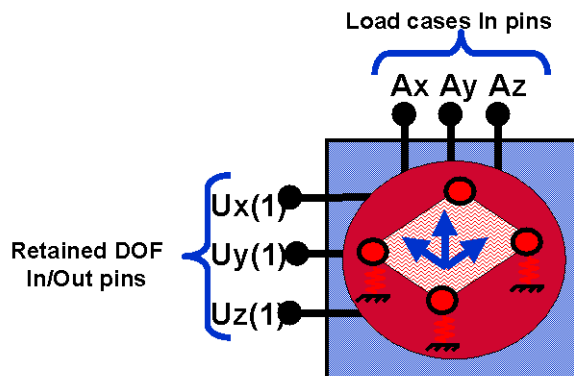


Figure 57: Model explanation

Load cases pins are input pins. You get as many input pins as the number of Load Steps you defined (in this case 3, for 3 possible different directions for the acceleration). You get as many output pins as the number of Master DOFs you defined (in this case 3, for 3 possible different directions of movement for the specified node (1))

There is no correlation between the number of output pins and the number of input pins.



### *Simulating a reduced model using the SPICE simulator*

Once you have created your SPICE model, you can simulate it using T-Spice.

- ☑ Launch **S-Edit** by selecting **Programs > Tanner MEMS Pro > S-Edit**.
- ☑ Click **File > Open** and browse for the *accel3.sdb* file.

The schematic view of the accelerotor appears in the S-Edit window (Figure 58).

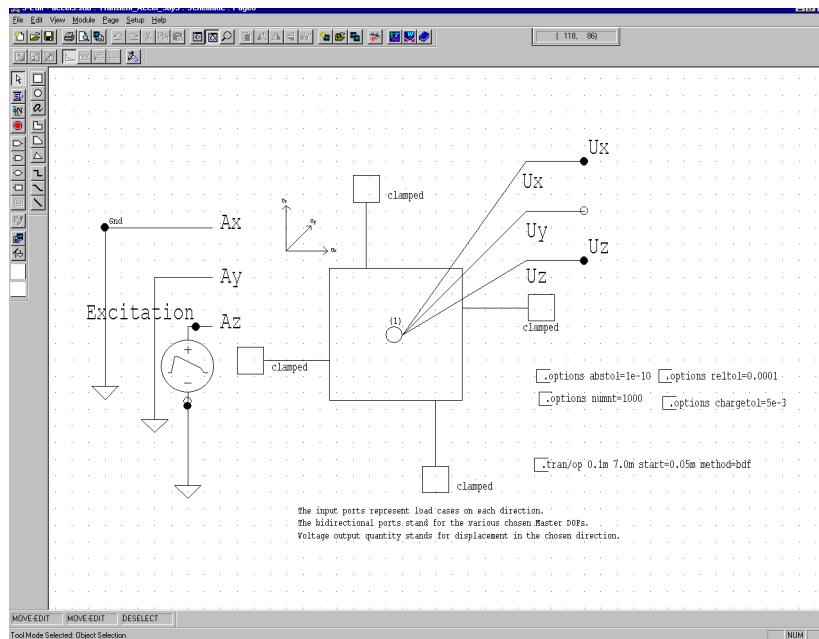


Figure 58: Schematic view of the accelerometer



Click **Module > Open**.

The **Open Module** window appears.

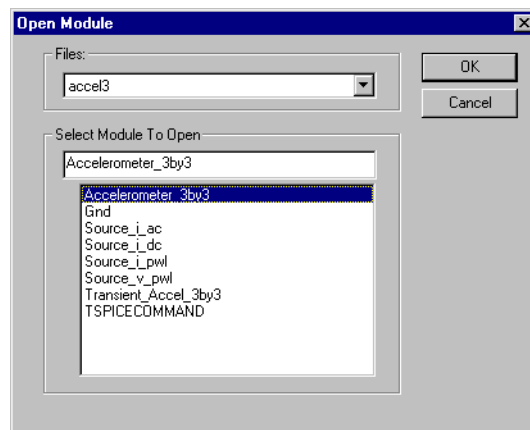


Figure 59: **Open Module** window

- ☒ Choose *Accelerometer\_3by3* as the **Module to Open** and click **OK**.

The module appears in the S-Edit window.

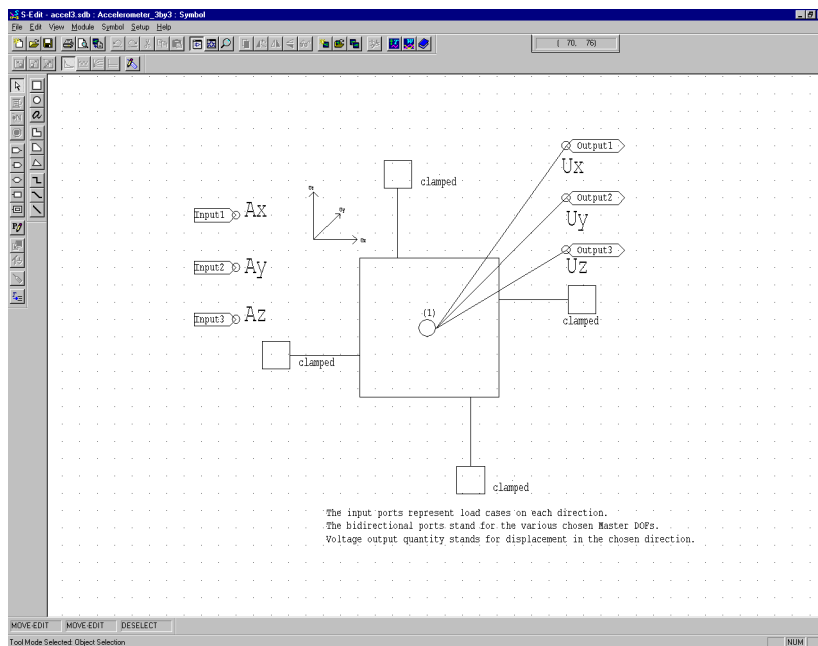


Figure 60: Schematic view of the *Accelerometer\_3by3* module

- ☒ Click **View > Schematic Mode**.
- ☒ Click the T-Space **Command Tool** (Figure 61).



Figure 61: T-Spice **Command Tool**

The T-Spice **Command Tool** window appears.

- ☒ Left-click anywhere in the blank design sheet.
- ☒ Select **Files > Include Files** and click **Browse** in the right part of the window.

The **T-Spice Command Tool** window appears.

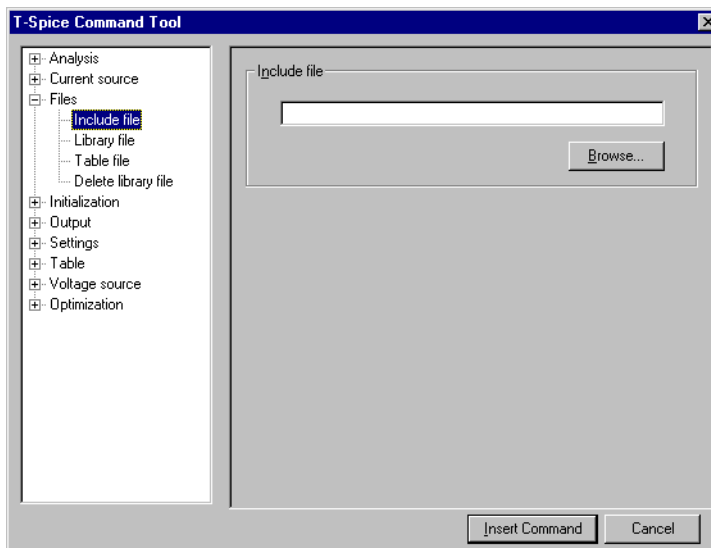


Figure 62: T-Spice Command Tool window

- ☑ Choose the previously created spice model (MyExample3.sp) and click **Open**.

If you did not follow the first part of the tutorial (generation of the SPICE model), use our spice model named *example3.sp* located in the **tutorial** directory.

- ☒ Click **Insert Command**.

A T-Spice command line that loads the generated model is then instantiated within the schematic view of the module.

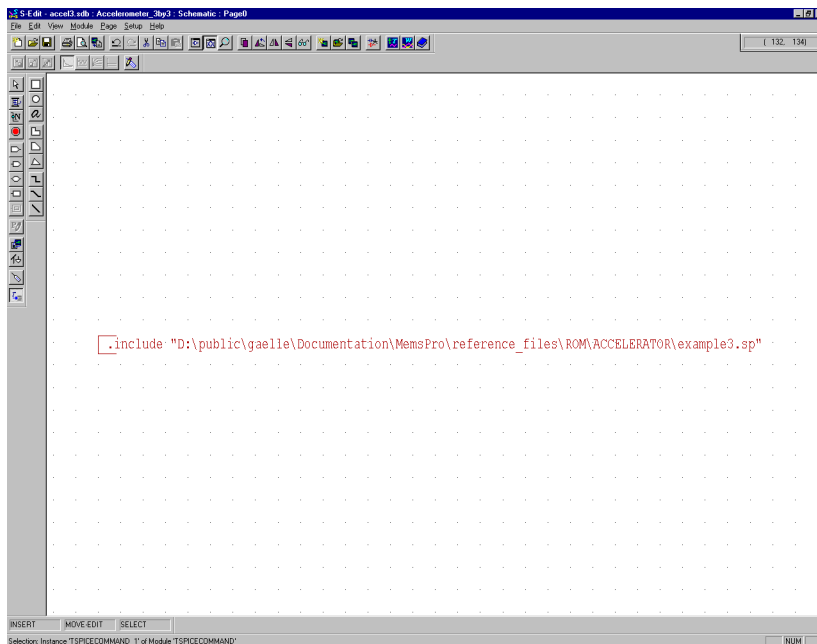


Figure 63: Viewing the command line

- ☑ Click **View > Symbol Mode**.

You can check that pin names match the names described in the model description.

- ☑ Click **Module > Open** and set *accel3* as **Files** and *Transient\_Accel\_3by3* as **Module to Open**.

- ☑ Click **OK**.

The new module opens in the S-Edit window.

- ☑ Select **Setup > Probing**.

The **Waveform Probing Setup** dialog box appears.



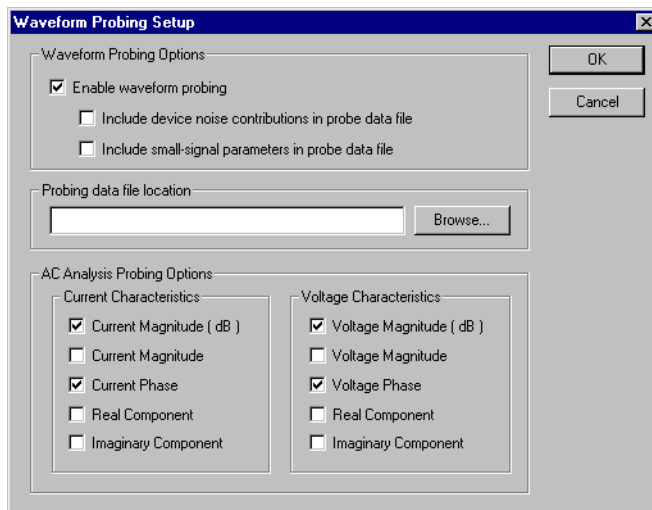


Figure 64: **Waveform Probing Setup** dialog box

- ☑ Click the **Browse** button and browse for the *accel3.dat* file.
- ☑ Click **Open** and then click **OK**.
- ☑ Click the **T-Spice** button.

An S-Edit warning message appears asking you whether you want to overwrite the existing file.

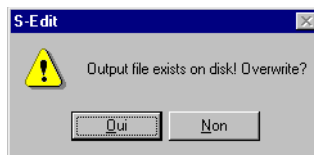


Figure 65: **S-Edit** warning message

- ☒ Click **Yes**.

The netlist generated by S-Edit shows up in the T-Spice window (Figure 66).

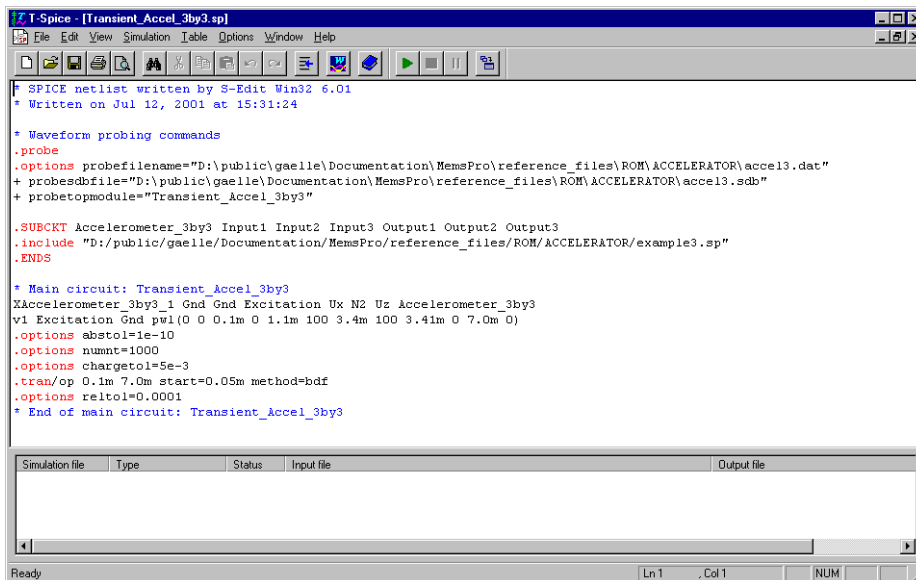


Figure 66: Viewing the generated netlist in T-Spice



Launch the simulation by clicking the **Run Simulation** button.

The **Run Simulation** dialog box appears.

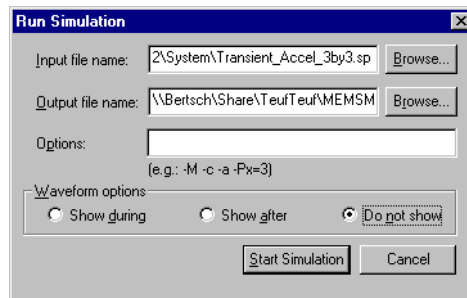


Figure 67: **Run Simulation** dialog box

☑ Check the **Do not show** box and click the **Start Simulation** button.

☑ Click **Yes** when asked if you want to overwrite the existing file.

A Simulation Output window opens presenting the results of the simulation.

☑ Click the **Probe** button.



Figure 68: **Probe** button

- ☒ Probe the **Az** node by clicking on it.

A **W-Edit** window opens. It contains the chart representing the result of the simulation performed on the Az node (Figure 69). This chart shows the excitation results.

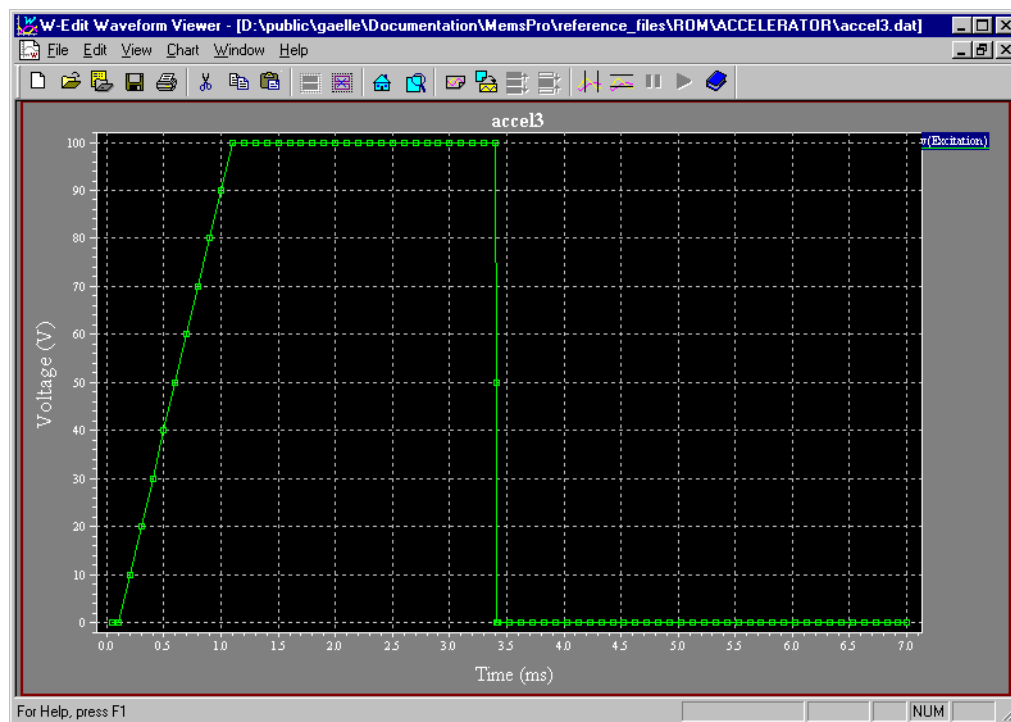


Figure 69: Viewing the results of the excitation simulation

- ☑ Access back the **S-Edit** window and probe for the **Uz** node.

A new chart appears in the **W-Edit** window.

- ☑ To view only the results of the last simulation, click **Chart > Traces** and unselect the excitation chart in the right part of the **Traces** dialog box.
- ☑ Then, select **Chart > Expand Chart** to obtain an expanded view of the chart (Figure 70).

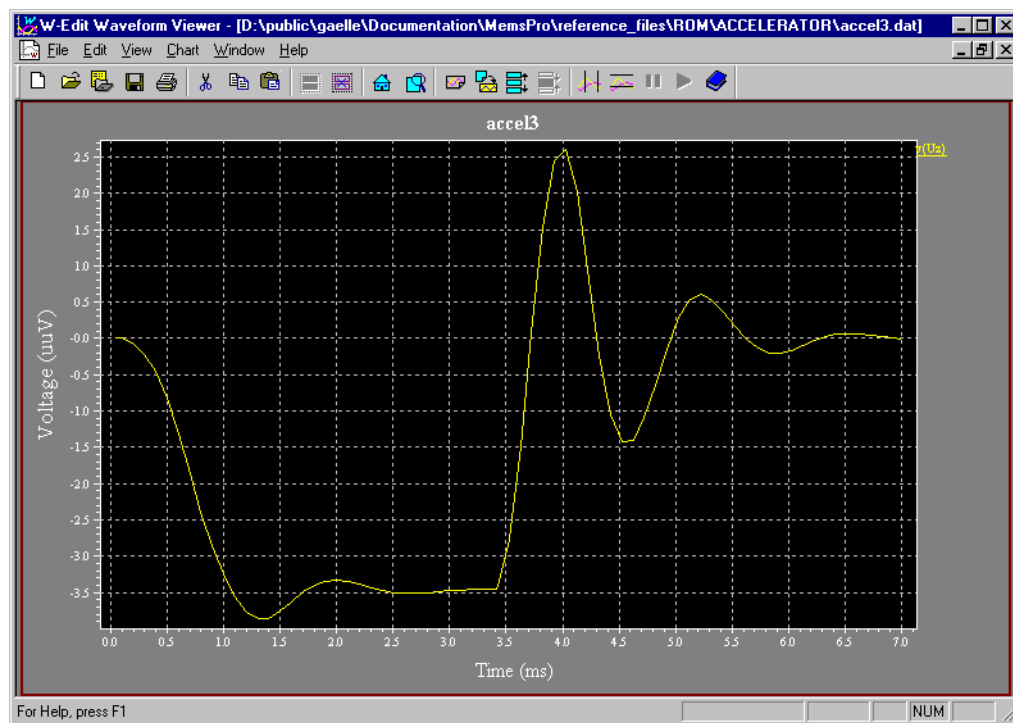


Figure 70: Viewing the results of the Uz simulation



## Reduction of Electrostatically Coupled Structural Systems

The electrostatic part of the R.O.M. tool gives access to an algorithm for the reduction of electrostatically coupled structural system. In this algorithm, the system can only be reduced in terms of a single structural degree of freedom.

Before using the R.O.M. tool, you first have to load in ANSYS a finite element model and provide all the data related to both electrical and structural behavior of your system.

The method used to provide a complete finite element model is the generation of a common geometrical model and the separate description of an electrical and structural environment, using “Physics Files”.

In this example, use the model of an electrostatically actuated micro mirror.

It consists of a plate connected at one end to a square beam that behaves as a torsional spring. The lower side of the plate is separated from the electrical ground by an air gap. The electrostatic interaction between the structure and the ground is assumed to be restrained to the space located under the plate and boundary effects are neglected. As the model is symmetric, only one part of the structure is modeled and appropriate boundary conditions applied to the symmetry plane.

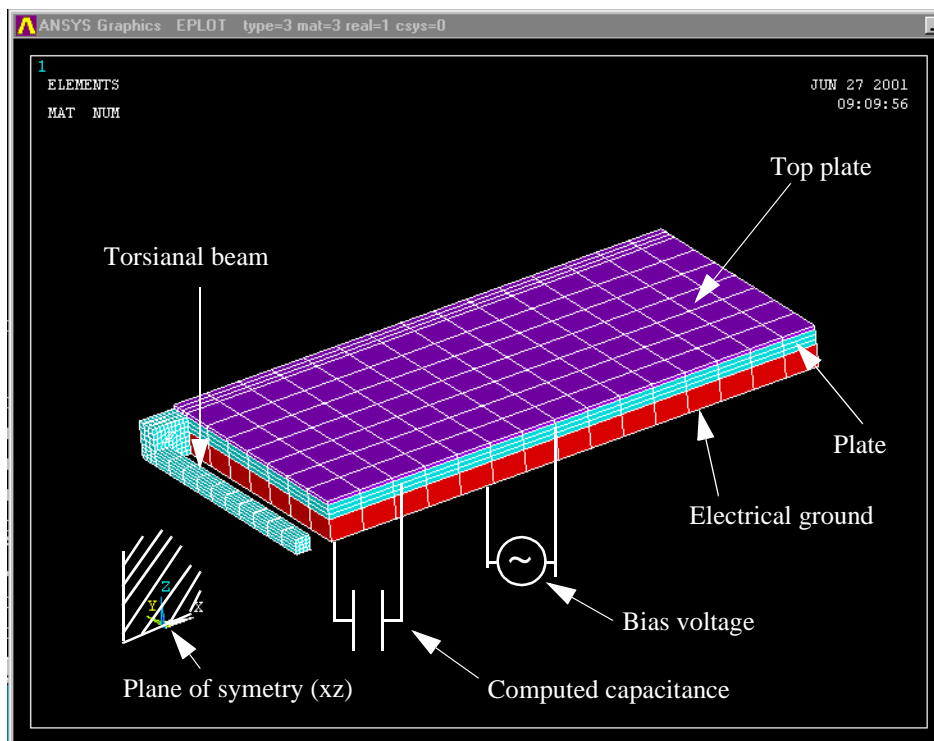


Figure 71: Electrostatically actuated micro mirror description

## Model Generation

First, provide a finite element model of your structure. In this example, run a macro. You can use other tools (the GUI, for example) to do this.

- ☒ Copy the following files to your working directory:

gen\_esman.macro, esman.mdl, esman.elec, esman.str.

They are all located under the same **tutorial** directory:

- ☒ In the **ANSYS Input** window, execute the macro that generates the model of the electrostatically actuated micro mirror and related “Physics Files”.

```
*USE,gen_esman.macro
```

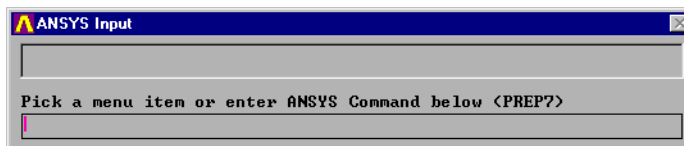


Figure 72: **ANSYS Input** window

The **ANSYS Graphics EPLLOT** window shows the 3D view of the model, and its meshing (Figure 71).

## Performing Reduction

You now have to define a single degree of freedom to which the model will be reduced.

In this example, you are interested in the model behavior at a particular node that has a number attributed to the `N_MASTER` variable (see the model description macro). It is the node located at the end of the top face, in the symmetry plane. In this example, you chose to use the vertical displacement as master degree of freedom.

- ☒ Set the degree of freedom to **Master Dof** by entering the following command in the **ANSYS Input** window (Figure 73):

`M,N_MASTER,UZ`

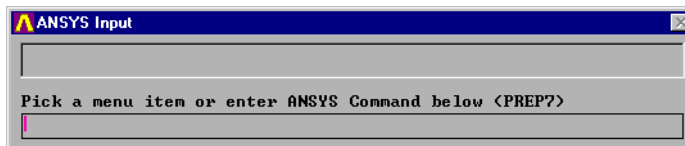


Figure 73: **ANSYS Input** window

- ☒ Click **MEMSCAP Tools > R.O.M Tools**.

The **R.O.M. Tools** menu appears (Figure 74).

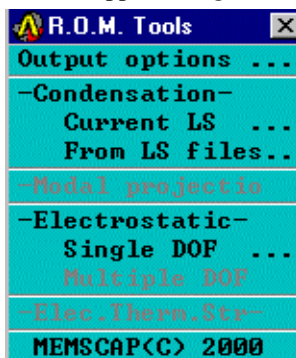


Figure 74: **R.O.M. Tools** menu

It gives access to all the condensation algorithms implemented in the MEMSCAP tool.

Before performing the reduction, you must select the format(s) in which the reduced models will be generated.

- ☒ Click **Output options**

The **Output options** (Figure 75) dialog box appears. It allows you to save your result file in HDLA™, SPICE or both languages.

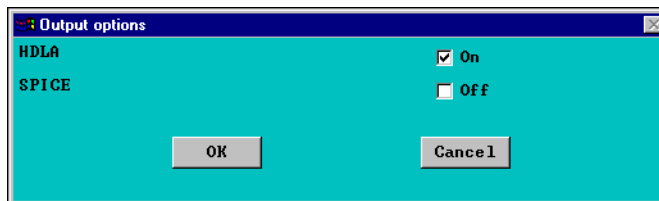


Figure 75: **Output options** dialog box

In this example, the SPICE language is chosen.

- ☑ Click **SPICE**.

Now, you may start the reduction algorithm.

- ☑ Click the **Single DOF** button, under the electrostatic title.

**Electrostatic-structure reduction to single DOF**

Enter output file radical: ''

- structure PHYSICS -

Title: ''

file radical: 'structu'

file extension: 'phy'

- electrostatic PHYSICS -

Title: ''

file radical: 'electric'

file extension: 'phy'

- Electrical parameters -

Compute capacitance: ☒ Yes

Conductors component name: 'cond'

Number of conductors: 3

Ground key: Last conductor

Excitation component name: 'cond2'

- algorithm parameters -

MINimal fitting value:

MAXimal fitting value:

Number of fitting points:

Degree of fitting: 4

Reference bias voltage: 1

MAXimum number of eigen modes: 0

OK Cancel

Figure 76: **Data** dialog box

The above dialog box opens, prompting you to define data and algorithm control parameters. For some parameters, default values are suggested, but you must check their compatibility with your own configuration.

- ☑ First, indicate the radical prefix of the file name used to generate the results. In this example, the name is “MyExample”.
- ☑ Under **structure PHYSICS** and **electrostatic PHYSICS**, accept the default parameters. These definitions must be in accordance with your generated files (see gen\_esman.macro).
- ☑ Then, define the **Electrical parameters**.

In this example, the relationship between the capacitance and the selected degree of freedom is modeled by selecting **Yes** in the appropriate field.

In this model, conductors are associated to components of nodes respectively called “cond1” and “cond2”. Then, the conductor name is “cond” and the number of conductors is 2. The “cond2” conductor is assumed to represent the electrical ground. The first conductor (“cond1”) is used to apply electrical excitation.

You now have to define the **algorithm parameters**.

In this example, to generate the reduced model, assume that the master degree of freedom varies between -2 microns and 1 micron. These values are the minimal and maximal fitting values.



In this range, take 10 points, and the resulting polynomial is of the 4th order.

The last two boxes keep their default values in this example. They are advanced parameters.

During the execution of the reduction algorithm, information is printed in the **ANSYS Output** window. The display is related to the algorithm status, results and accuracy estimation.

The first information displayed is the title of the algorithm and the summary of the data you entered in the previous dialog box.

```
*****
Electrostatic reduction (single degree of freedom)
*****

Structure PHYSICS:
File    = "structu.phy"
Title   = ""

Electrostatic PHYSICS:
File    = "electric.phy"
Title   = ""

Conductors name           = "cond"
Number of conductors      = 2
Ground key                 = 0
Capacitance matrix dimension = 1
Excitation component      = "cond2"
```

The next display is related to the structural condensation algorithm. The name of the ANSYS substructure file is printed as well as the reduced values of mass, damping and stiffness.

```
=====
Run structural substructuring
=====

Ansys substructure file = "tmp.sub"

Waiting for ANSYS solution ... Done

Structural reduced parameters:

Stiffness(k) = 1.1646848e+00
Mass      (m) = 4.9866184e-12
Damping   (c) = 0.0000000e+00
```

Then comes the modal analysis and the frequency response tuning procedure. Structural eigen frequencies computed with the complete model are printed, followed by a comparison between the estimated eigen frequencies and mass values.

The comparison of the values gives an indication on the representation of the mode by the selected master degree of freedom.

```
=====
Run structural Modal analysis
=====
```

Waiting for ANSYS solution ... Done

```
-----
Mode      Frequency
-----
      1      7.6670774e+04
-----
```

Tuning transient response:

Expected	eigen frequency = 7.6670774e+04
Approximated	eigen frequency = 7.6916812e+04 (0.3209 % shift)
Reduced	mass = 4.9866184e-12
Corrected	mass = 5.0186740e-12 (0.6428 % shift)

The next output indicates the sweeping parameters for coupled effects evaluation. A status is printed after each set of analyses.

```
=====
Perform coupled analyses
=====
```

```
Number of analyses      = 10
Degree of fitting       = 4
Minimum DOF value       = -2.0000000e-06
Maximum DOF value       = 1.0000000e-06
Reference bias voltage   = 1.0000000e+00
```

```

--> Performing set of coupled analyses number 1/10
--> Performing set of coupled analyses number 2/10
--> Performing set of coupled analyses number 3/10
--> Performing set of coupled analyses number 4/10
--> Performing set of coupled analyses number 5/10
--> Performing set of coupled analyses number 6/10
--> Performing set of coupled analyses number 7/10
--> Performing set of coupled analyses number 8/10
--> Performing set of coupled analyses number 9/10
--> Performing set of coupled analyses number 10/10

```

At the end of the coupled effect evaluations, a summary of analysis points is printed.

As a single conductor is modeled, there is only one capacitance value, between this conductor and ground.

Point	Master DOF	Reduced FMAG	Capa[1,1]
1	-2.0000000e-06	-2.5765530e-09	1.2358000e-14
2	-1.6666667e-06	-1.8065974e-09	1.0862294e-14
3	-1.3333333e-06	-1.3646935e-09	9.7759145e-15
4	-1.0000000e-06	-1.0811644e-09	8.9370341e-15
5	-6.6666667e-07	-8.8548790e-10	8.2623515e-15
6	-3.3333333e-07	-7.4327955e-10	7.7036706e-15
7	4.2351647e-22	-6.3585629e-10	7.2307667e-15
8	3.3333333e-07	-5.5223359e-10	6.8235294e-15
9	6.6666667e-07	-4.8555243e-10	6.4679561e-15
10	1.0000000e-06	-4.3132108e-10	6.1539385e-15

Once the numerical values of reduced forces and capacitance (if requested) have been approximated by an analytical expression, an accuracy estimation of the fitting process is performed on each fitted value.

This evaluation consists in the comparison, at each fitting point, of the numerical value with the analytical expression.

A summary of the greatest absolute difference is also printed and compared with the absolute mean value of the numerical values in order to obtain a relative data.

```
=====
Process coupled analyses results
=====
```

FMAG accuracy estimation:

Point	Reference	Approximation	Difference
1	-2.5765530e-09	-2.5766599e-09	1.0695196e-13
2	-1.8065974e-09	-1.8064695e-09	-1.2799081e-13
3	-1.3646935e-09	-1.3647061e-09	1.2506981e-14
4	-1.0811644e-09	-1.0811986e-09	3.4230521e-14
5	-8.8548790e-10	-8.8549709e-10	9.1917686e-15
6	-7.4327955e-10	-7.4326989e-10	-9.6567069e-15
7	-6.3585629e-10	-6.3584617e-10	-1.0119246e-14
8	-5.5223359e-10	-5.5223378e-10	1.8783152e-16
9	-4.8555243e-10	-4.8555978e-10	7.3576808e-15
10	-4.3132108e-10	-4.3131849e-10	-2.5912165e-15

```

Mean      absolute value      = 1.0562739e-09
Maximum absolute difference    = 1.2799081e-13 (0.0121 %)

```

Capa [1,1] accuracy estimation:

Point	Reference	Approximation	Difference
1	1.2358000e-14	1.2356427e-14	1.5729220e-18
2	1.0862294e-14	1.0865342e-14	-3.0473488e-18
3	9.7759145e-15	9.7753145e-15	5.9910553e-19
4	8.9370341e-15	8.9355797e-15	1.4543140e-18
5	8.2623515e-15	8.2619657e-15	3.8578644e-19
6	7.7036706e-15	7.7043433e-15	-6.7263844e-19
7	7.2307667e-15	7.2315074e-15	-7.4077217e-19
8	6.8235294e-15	6.8234690e-15	6.0393615e-20
9	6.4679561e-15	6.4672164e-15	7.3966534e-19
10	6.1539385e-15	6.1542450e-15	-3.0649302e-19

```

Mean      absolute value      = 8.4575456e-15
Maximum absolute difference    = 3.0473488e-18 (0.0360 %)

```

```

*****
Reduced model successfully generated !
*****

```

After execution, results files are created in your working directory. They contain the behavioral model, written in the selected format. In this case, the output file is MyExample.sp.

### *Simulating a reduced model using the SPICE simulator*

Once you have created your SPICE model, you can simulate it using T-Spice.

- ☑ Launch **S-Edit** by selecting **Programs > Tanner MEMS Pro > S-Edit**.
- ☑ Click **File > Open** and browse for the *MicroMirror.sdb* file.

The schematic view of the micro mirror appears in the S-Edit window (Figure 77).

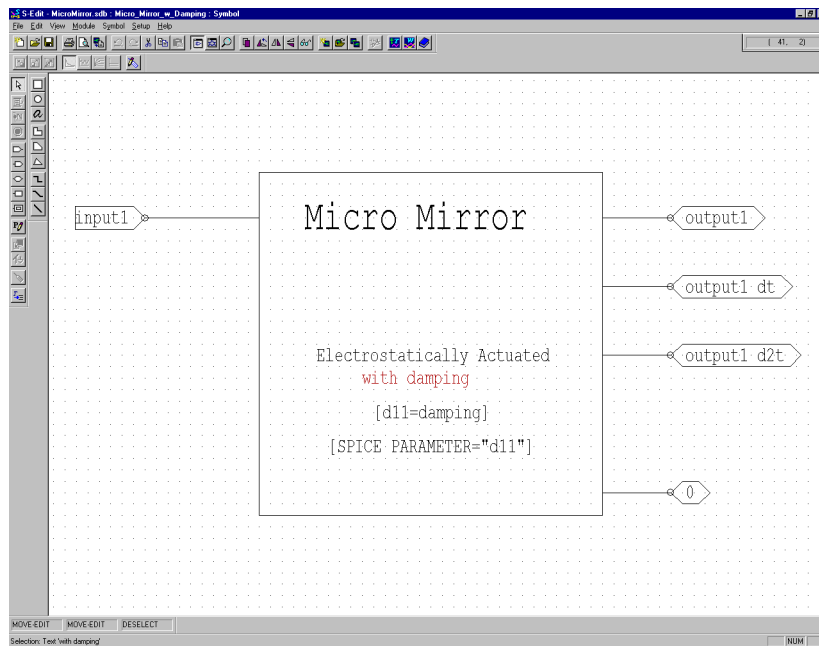


Figure 77: Schematic view of the micro mirror

- ☒ Click **Module > Open**.



The **Open Module** window appears .

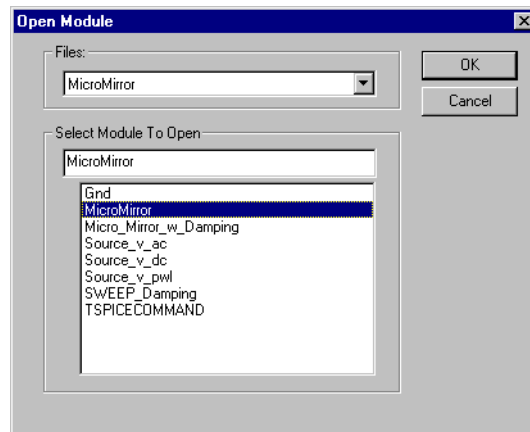


Figure 78: **Open Module** window

- ☒ Choose MicroMirror and click **OK**.
- ☒ Click **View > Schematic Mode**.
- ☒ Click the T-Spice **Command Tool**.



Figure 79: T-Spice **Command Tool**

- ☒ Click somewhere in the blank window.

The T-Spice **Command Tool** window appears.

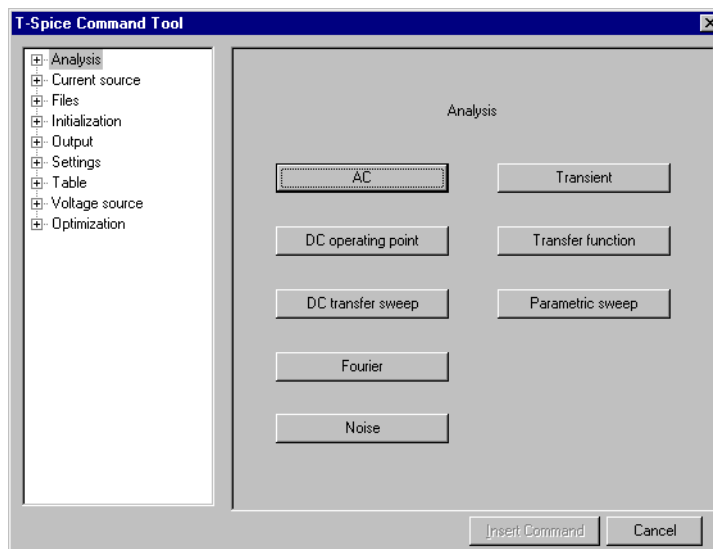


Figure 80: **T-Spice Command Tool** window

- ☑ Left-click anywhere in the blank design sheet.
- ☑ Select **Files > Include file** and click **Browse** in the right part of the window.
- ☑ Choose the previously created SPICE model (MyExample.sp) and click **Open**.

If you did not follow the first part of the tutorial (generation of the spice model), use our SPICE model named *example.sp* located in the **tutorial** directory.

- ☑ Click **Insert Command**.

A T-Spice command line that loads the generated model is then instantiated within the schematic view of the module.

- ☑ Click **View > Symbol Mode**.

You can check that the pin names match the names described in the model description of the previous part. In this case, the ground pin name is 0.

We created the symbol view of the device. It can be re-used as such since the generated model uses a fixed template. We added some commands to declare d11 as the damping parameter we want to sweep.

Now, you will study the influence of damping on the chosen device.

You will perform a step excitation and look at the response of the device with various damping parameters.

You will sweep the damping parameter between typical values for MEMS:  $1\text{e-}7$  to  $1\text{e-}3 \text{ N.s.m}^{-1}$ .

- ☑ Select **Module > Open**.

- ☒ Choose ***SWEEP\_Damping*** in the **Open Module** dialog box.

The appropriate module appears in the S-Edit window.

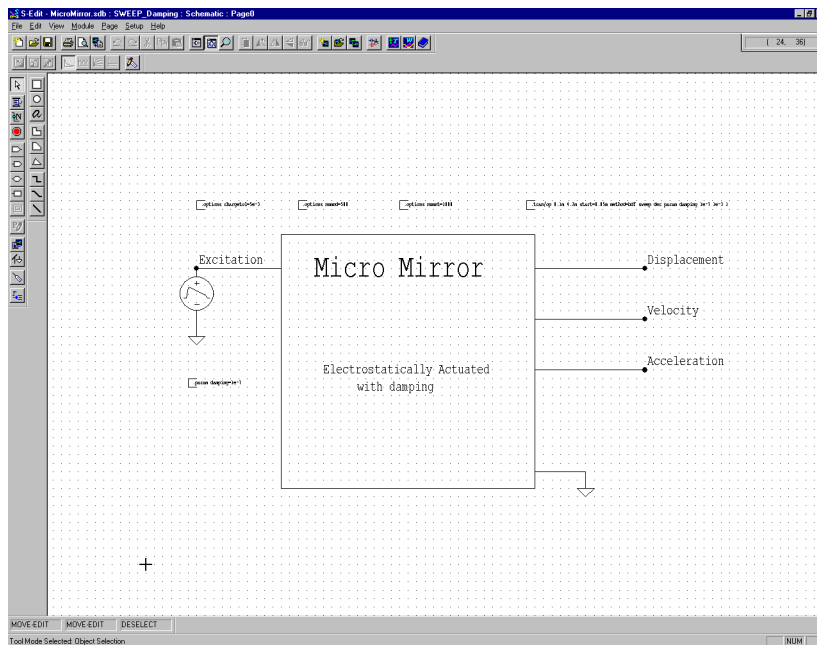


Figure 81: Viewing the SWEEP\_Damping module

- ☑ Select **Setup > Probing**.

The **Waveform Probing Setup** dialog box appears.

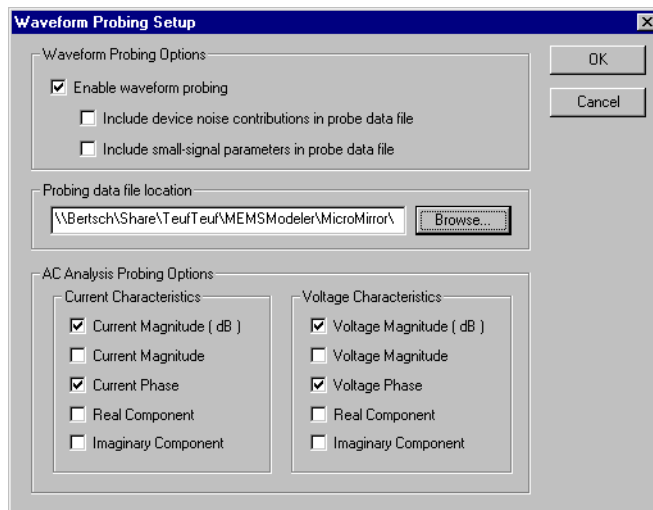


Figure 82: **Waveform Probing Setup** dialog box

- ☑ Click the **Browse** button and browse for the *MicroMirror.dat* file located in the tutorial directory.

- ☑ Click **Open** and then click **OK**.
- ☑ Click the **T-Spice** button.

An S-Edit warning appears asking you whether you want to overwrite the existing file or not.

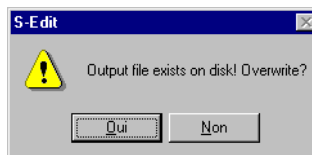
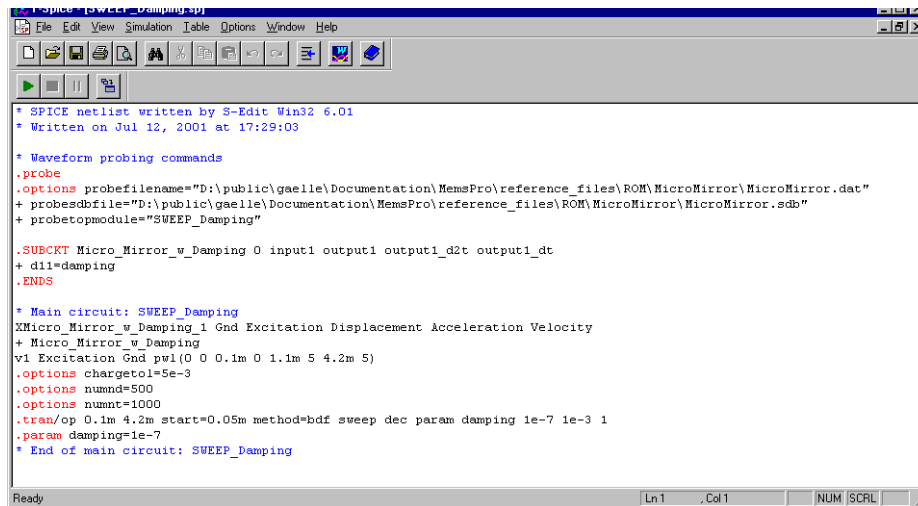


Figure 83: S-Edit warning

- ☑ Click **Yes**.

The netlist generated by S-Edit opens in the T-Spice window.



```

* SPICE netlist written by S-Edit Win32 6.01
* Written on Jul 12, 2001 at 17:29:03

* Waveform probing commands
.probe
.options probefilename="D:\public\gaelle\Documentation\MemsPro\reference_files\ROM\MicroMirror\MicroMirror.dat"
+ probesdbfile="D:\public\gaelle\Documentation\MemsPro\reference_files\ROM\MicroMirror\MicroMirror.sdb"
+ probetopmodule="SWEEP_Damping"

.SUBCKT Micro_Mirror_w_Damping 0 input1 output1 output1_d2t output1_dt
+ d11=damping
.ENDS

* Main circuit: SWEEP_Damping
XMicro_Mirror_w_Damping 1 Gnd Excitation Displacement Acceleration Velocity
+ Micro_Mirror_w_Damping
v1 Excitation Gnd pwl(0 0 0.1m 0 1.1m 5 4.2m 5)
.options chargetol=5e-3
.options nummd=500
.options numnt=1000
.tran/op 0.1m 4.2m start=0.05m method=bdf sweep dec param damping 1e-7 1e-3 1
.param damping=1e-7
* End of main circuit: SWEEP_Damping

```

Figure 84: Viewing the generated netlist

- ☒ Launch the simulation by clicking the **Run Simulation** button.

The **Run Simulation** dialog box opens.



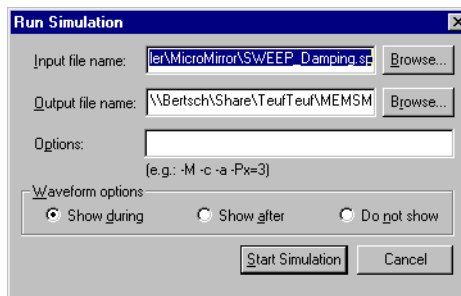


Figure 85: Run Simulation dialog

- ☑ Check the **Do not Show** box and click the **Start Simulation** button.
- ☑ Click **Yes** when asked if you want to overwrite the existing file.
- ☑ Access back the S-Edit window and probe for the node called displacement.

The chart corresponding to the displacement results appears in the W-Edit window.

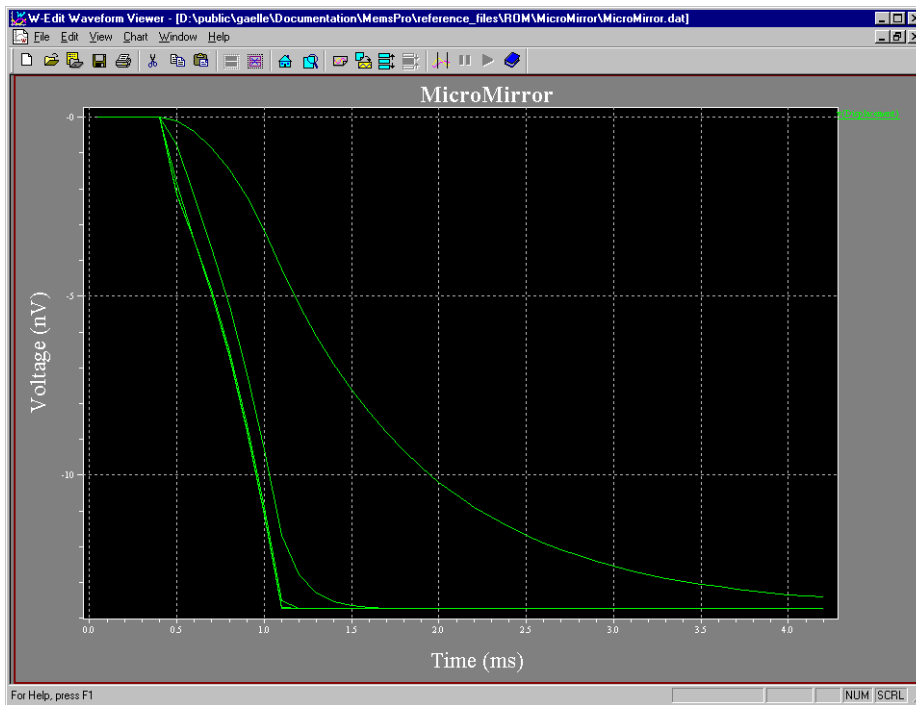


Figure 86: Viewing the 5 displacement values

- ☒ In the W-Edit window, select **Chart > Traces**.

- ☒ Keep only the first and last traces (corresponding to the extreme values for the damping coefficient) by unchecking the other boxes.
- ☒ Click **OK**.

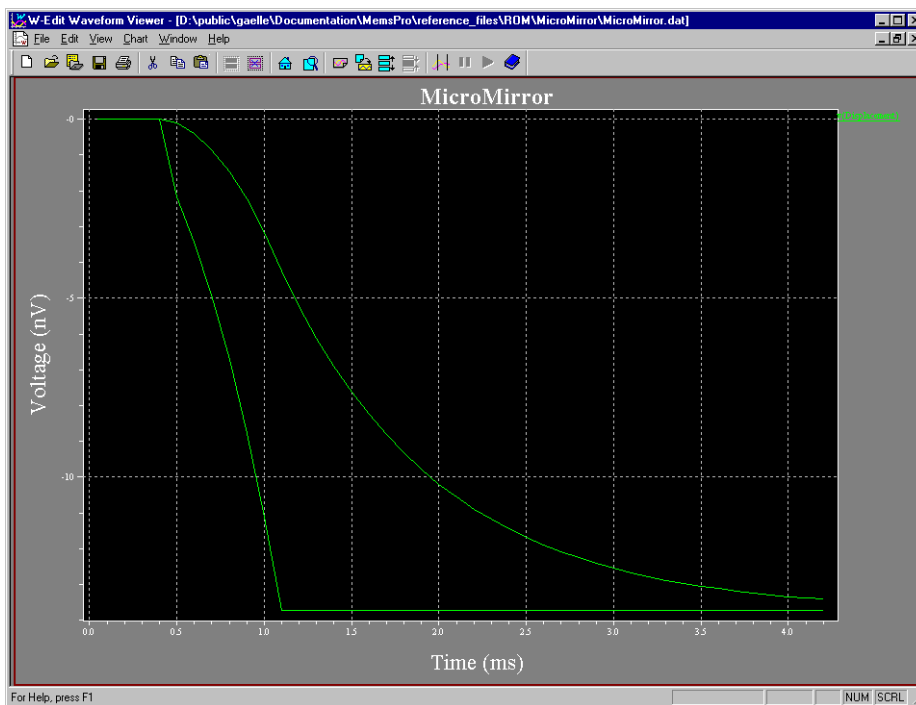


Figure 87: Viewing the extreme values for the displacement

- ☒ You can now close S-Edit.

# 10 Optimization Tutorial

---

▪ Introduction	437
▪ Setting up the Optimization	439
▪ Running the Optimization	453
▪ Examining the Output	454

# Introduction

Optimization is a critical tool for the MEMS engineer. The MEMS Pro optimization engine lets you tune the parameters of your system to achieve its best possible performance. Optimization is achieved by running iterative simulations over a constrained set of selected parameters. In order to specify an optimization, you must supply a list of parameters, the optimization goal or goals, and your choice among the analysis and the optimization algorithms we provide. Further, you decide which measurements will be used to determine if the optimization has been successful.

Once you have successfully run an optimization, the optimized parameter values can be used in subsequent analyses of the same model. This allows for incremental optimization: some parameters can be optimized while others are held fixed; later, other parameters can be optimized based on the results of the earlier optimization. If multiple analyses are requested, DC analyses will be performed first, then AC analyses and then transient analyses. Multiple analyses of the same type are performed in the order they appear in the input file.

The optimization process is most easily explained by walking through some examples. Our first example is a simple optimization with just one parameter, measure, analysis and goal.

**Note**

---

For more information on *optimizing your design*, see Optimization on page 130 of the *T-Spice Pro User Guide*.

---

## Setting up the Optimization

If you recall the MEMS Pro Tutorial on page 14, we explored the construction and behavior of a resonator. Here, we explore that model further. The MEMS Pro optimizer will help you find the value of the **springlength** parameter that will most closely achieve the optimization goal: a resonant frequency of 40 kHz.



Launch S-Edit by double-clicking the **S-Edit** icon





- ☑ To open the tutorial, select **File > Open** and choose the file called **<install directory>\Examples\optimize\resonator\reson.sdb**. The schematic appears in the S-Edit window (Figure 88).

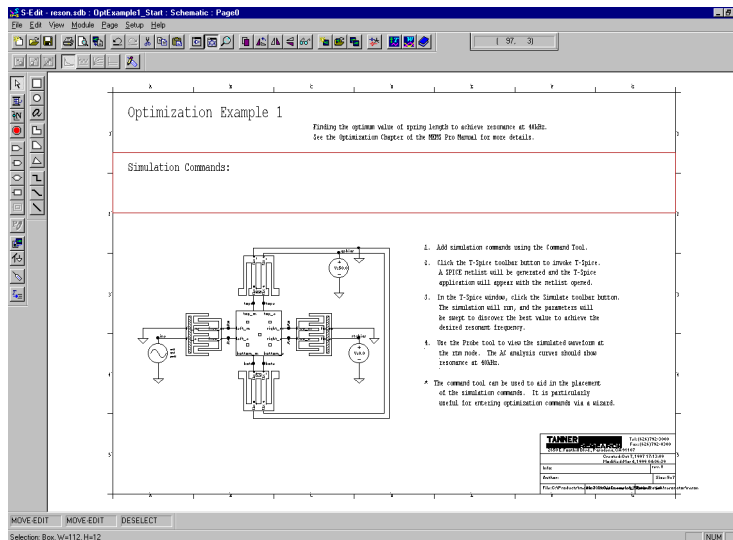


Figure 88: Viewing the resonator in S-Edit

Now, we need to associate process and material properties to the model.

- ☑ Select the **Command Tool** button to enter the **Command** tool mode, or, left-click on the work area to invoke the **T-Spice Command Tool** dialog (Figure 89).

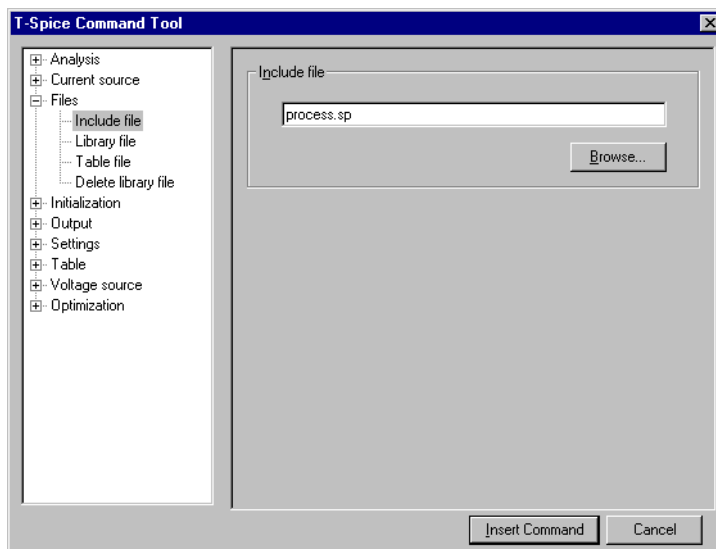


Figure 89: **T-Spice Command Tool** dialog

- ☑ In the left-hand tree, double-click **Files** and then **Include file**. Type *process.sp* in the **Include file** field. Click **Insert Command**

The optimization engine needs to know what analysis we will use to determine whether we have reached our optimization goal.

- ☑ Click somewhere in the work area to invoke the **T-Spice Command Tool** dialog again.
- ☑ In the left-hand tree, double-click **Analysis** and then **AC**.
- ☑ Choose **decade** for **Frequency sampling type**, set **Frequencies per decade** to **500**, **Frequency range From** to **10k** and **Frequency Range To** to **100k** (Figure 90).
- ☑ Click **Insert Command**.

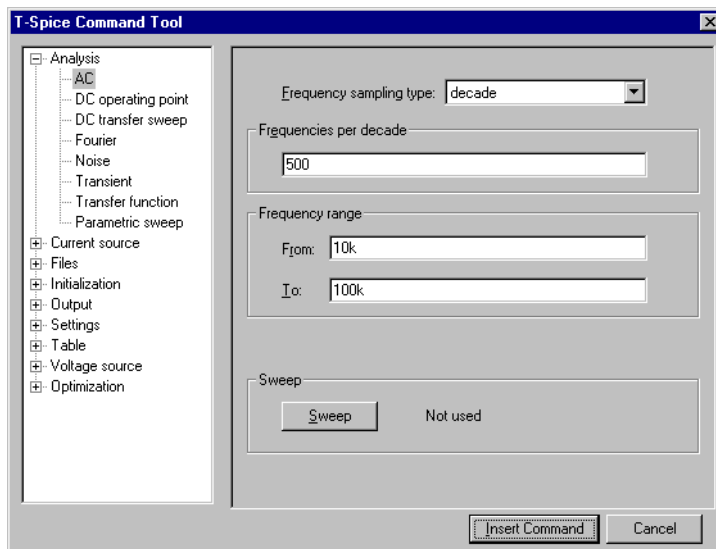


Figure 90: Customizing the AC analysis

Next, we need to select the parameters of interest in our model.

- ☑ Left-click into the work area to invoke the **T-Spice Command Tool** dialog again.

- ☑ In the left tree, double-click **Settings** and then **Parameters**. Add the **springlength** parameter statement. Set **Parameter name** to *springlength*, and **Parameter value** to *100e-6* (Figure 91). Click **Add**. Click **Insert Command**.

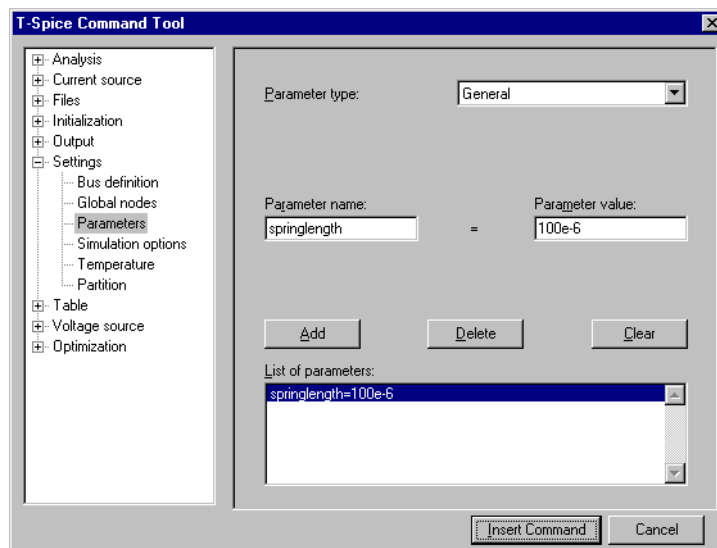


Figure 91: Customizing the setting parameters

Now, we define the quantities that will be measured during the simulation.

- ☑ Left-click into the work area to invoke the **T-Spice Command Tool** dialog again.
- ☑ In the left tree, double-click **Output** and then **Measure**. Set **Analysis type** to **AC** and **Measurement type** to **Find-when**. Enter **res\_freq** into the **Measurement result name** field. Under **Find**, click the **x-value** radio button. Under **When**, click the **Signal** radio button, and enter **vp(rtm)**. Set **equals value** to **90**. From the drop-down menu next to **on** select **crossing**. For **number**, select **1** (Figure 92). Click **Insert Command**.

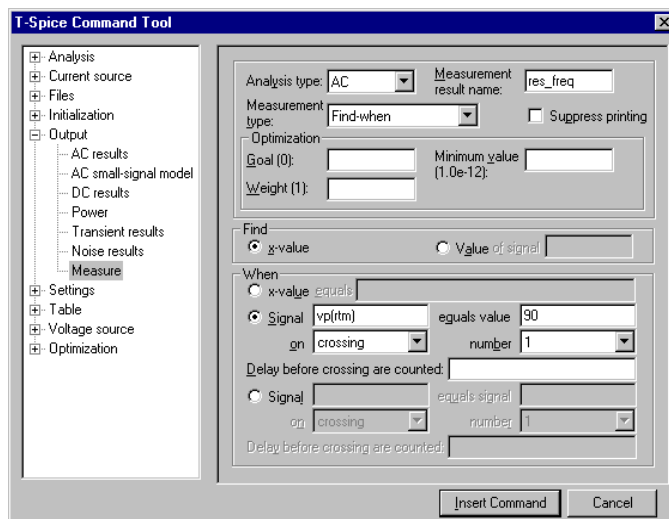


Figure 92: Customizing the quantities to measure

Now that we have set up the model, we are ready to set up the optimization.

- ☑ Left-click into the work area to invoke the **T-Spice Command Tool** dialog.
- ☑ In the left-hand tree, double-click **Optimization** (Figure 93).

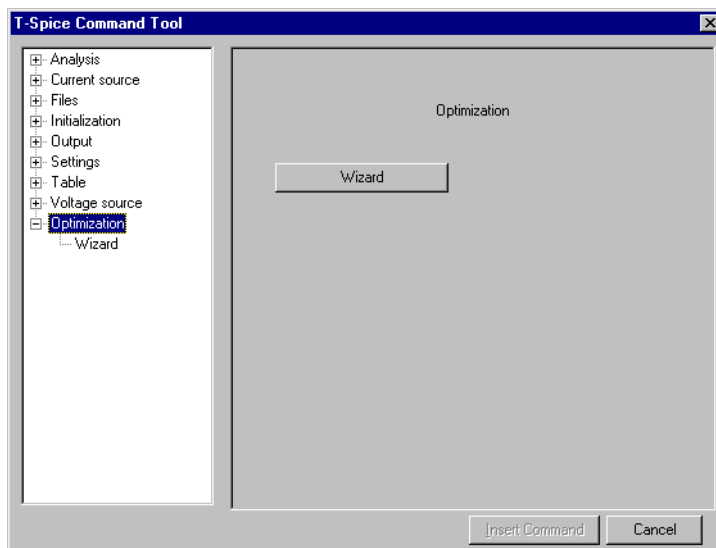


Figure 93: Customizing the optimization

- ☑ Click **Wizard** in the left tree or the **Wizard** button on the right to bring up the **Optimization setup** dialog.
- ☑ Enter **opt1** in the **Optimization name** field and type or select **First AC Analysis** as the **Analysis name** (Figure 94).

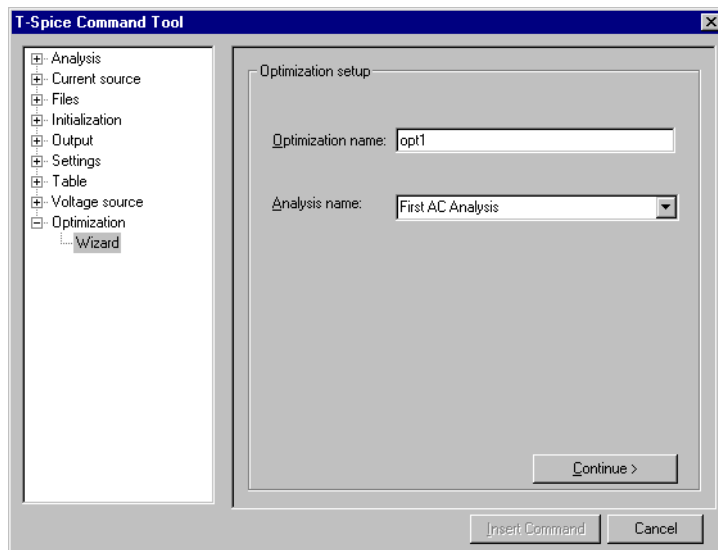


Figure 94: Customizing the optimization setup



- ☑ Click **Continue** to access the next dialog, **Set optimization goals**.
- ☑ Set **Measurement** to *res\_freq* and **Target value** to **40e3**. Click **Add** to add these values to the **List of optimization goals** (Figure 95). When you finish, the **T-Spice Command** tool dialog will look like the following:

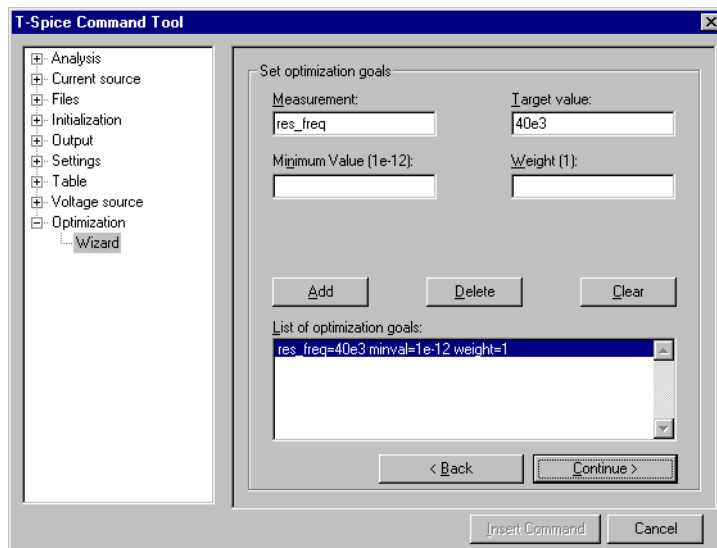


Figure 95: Customizing the optimization goals

Note that this target value will overwrite the one set earlier during the measurement setup.

- ☑ Click **Continue** to go to the next dialog, **Set parameter limits**.
- ☑ Set the **Parameter name** to *springlength*, **Minimum value** to *10e-6*, **Maximum value** to *200e-6*, **Delta (Optional)** to *0.25e-6*, and **Guess value (Optional)** to *100e-6* (Figure 96). Click **Add** to add the values to the **List of parameters**.

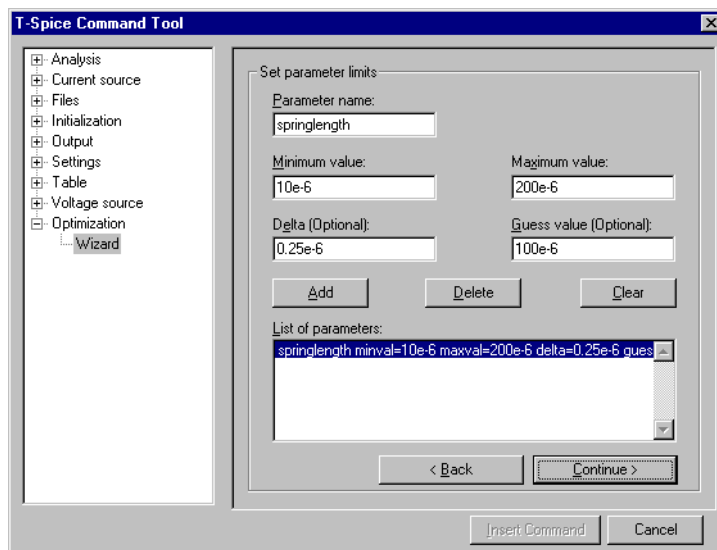


Figure 96: Customizing the parameters limits

- ☑ Click **Continue** to go to the next dialog, **Set optimization algorithm**.
- ☑ In the **Name** field, type **optmod**. For all other values, accept the defaults (Figure 97). When you are finished, the **T-Spice Command** tool dialog will look like the following:

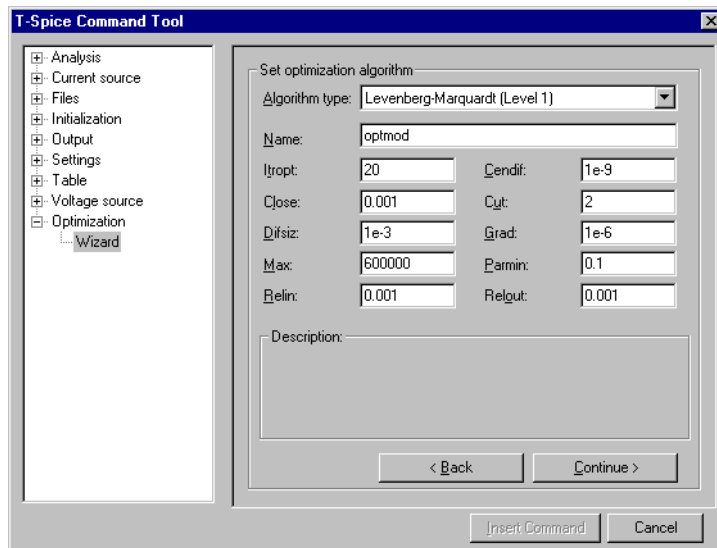


Figure 97: Customizing the optimization algorithm

- ☑ Click **Continue** to go to the next dialog, **Insert command**.
- ☑ The optimization commands you have created are displayed in the dialog. Review your entries; make sure they are correct. If you need to change a line, click **Back** to make changes. If the commands are correct, click **Insert Command**.

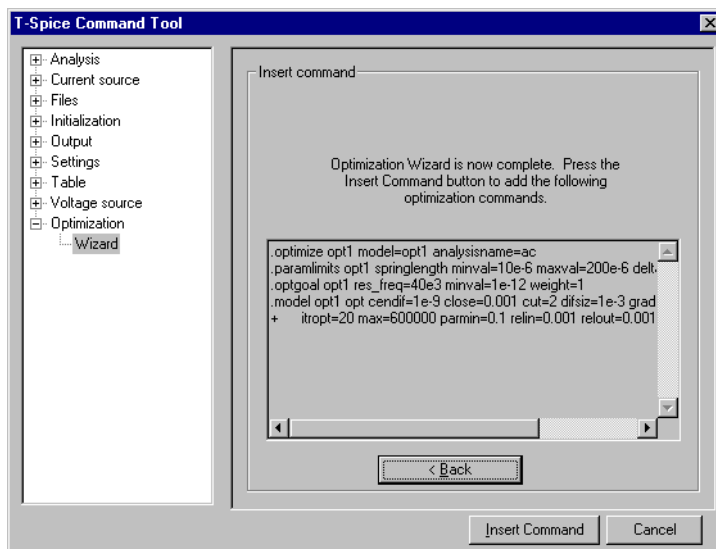



Figure 98: Finalizing the optimization customization



## Running the Optimization

- ☑ Clicking the **T-Spice** icon  located in the toolbar will launch T-Spice with the exported netlist open.
- ☑ Run the optimization by selecting **Simulation > Run Simulation**.

## Examining the Output

### OPTIMAL VALUES



The optimization engine output file contains the results of each simulation iteration. The value of each optimization parameter (submitted for a given run) appears followed by the gradient of the objective function at that parameter value. In the example, we had just one parameter, **springlength**. The next line contains the residual, or the difference between the output measure and your goal. The Levenberg-Marquardt algorithm is used for the optimization; the Marquardt value is an artifact of that algorithm.

Once the optimization engine produces a result that falls within the tolerance you have set, it desists. The final parameter estimate, **optimized parameter values: springlength = 1.1775e-004** and the goal measure **Measurement result summary - OPTIMIZE=opt1 res\_freq = 3.9922e+004** are written to the output file.

The output file for the optimization set up in our first example appears below.

```
Optimization parameters:
springlength = 0.0001                derivative = -3.73793
Optimization initialization: resid=0.278656 grad=3.73793
Marquardt=0.001
Optimization parameters:
springlength = 0.00011275            derivative = -0.640763
```

Optimization iteration 1: resid=0.067675 grad=0.640763  
Marquardt=0.0005  
Optimization parameters:  
springlength = 0.00011725 derivative = -0.0522022

Optimization iteration 2: resid=0.00669069 grad=0.0522022  
Marquardt=0.00025  
Optimization parameters:  
springlength = 0.0001175 derivative = -0.00137331

Optimization iteration 3: resid=0.00221041 grad=0.00137331  
Marquardt=0.000125

Optimization iteration 4: resid=0.00221041 grad=0.00137331  
Marquardt=0.0005

Optimization iteration 5: resid=0.00221041 grad=0.00137331  
Marquardt=0.002

Optimization iteration 6: resid=0.00221041 grad=0.00137331  
Marquardt=0.008

Optimization iteration 7: resid=0.00221041 grad=0.00137331  
Marquardt=0.032

Optimization iteration 8: resid=0.00221041 grad=0.00137331  
Marquardt=0.128



Optimization iteration 9: resid=0.00221041 grad=0.00137331  
Marquardt=0.512

Optimization iteration 10: resid=0.00221041 grad=0.00137331  
Marquardt=2.048

Optimization parameters:  
springlength = 0.00011775                      derivative = 0.00389856

Optimization iteration 11: resid=0.00195898 grad=0.00389856  
Marquardt=1.024

Optimization iteration 12: resid=0.00195898 grad=0.00389856  
Marquardt=4.096

Optimization parameters:  
springlength = 0.00011775                      derivative = 0.00389856

Optimization iteration 13: resid=0.00195898 grad=0.00389856  
Marquardt=2.048

**Optimized parameter values:**  
**springlength = 1.1775e-004**

**Measurement result summary - OPTIMIZE=opt1**  
**res\_freq = 3.9922e+004**

# 11 Verification

---

▪ Introduction	458
▪ Adding Connection Ports	459
▪ Extracting Layout	463
▪ Extracting Schematic for LVS	468
▪ Comparing Netlists	471

# Introduction

This chapter explains how to verify a mixed technology layout by showing the processes of layout extraction and layout vs. schematic comparison. This chapter contains a tutorial on these important design steps. The tutorial is a continuation of the main MEMS Pro tutorial from Chapter 2 - MEMS Pro Tutorial.

## Adding Connection Ports

We will begin the tutorial with the design you completed in Chapter 2 - MEMS Pro Tutorial. Please, open it in L-Edit.

- ☑ Launch L-Edit by double clicking the **L-Edit** icon and select **File > Open** to load the design file we have provided you, **reson.tdb**.

As long as geometrical objects on the same layers touch or overlap, they will be fabricated as connected, however, for SPICE netlist extraction to work correctly, the connection must be explicitly stated. These connections are called ports, and they define connectivity for a cell. Ports are objects drawn with the **Port** tool on a layer used specifically for interconnection.

### Note

---

For more information on *connecting ports*, see Drawing Objects on page 1-246, and Subcircuit Recognition on page 3-73 of the *L-Edit User Guide*.

---

Ports allow L-Edit's **Extract** command to recognize connectivity at a block or circuit level. The cells that were generated automatically for this design (the **plate**, **comb-drive**, **folded spring**, **ground plate**, and **bonding pad**) already have properly drawn ports in place. You can examine the port connections in the example **Resonator** by choosing **Edit > Find** to find objects of type port.

The location of ports in each of the layout cells is described and illustrated in Figure 99.

- The **plate** has 4 ports (**PL\_Left**, **PL\_Right**, **PL\_Top**, **PL\_Bottom**). They look like long rectangles (2 units thick) stretching across the length of each of the four sides of the plate.

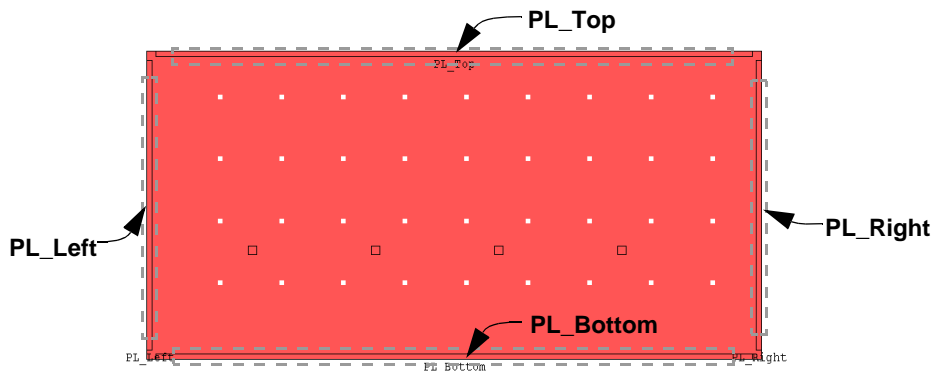



Figure 99: Ports of the plate element

- The **comb-drive** has 2 ports (**C\_Free**, **C\_Fixed**). They look like long rectangles (2 units thick) stretching across the left and right sides of the element.

- The **folded spring** has 2 ports (**FS\_Free**, **FS\_Fixed**). **FS\_Free** looks like a rectangle stretching across the bottom of the element. **FS\_Fixed** looks like a rectangle attached to the right side of the element overlapping the anchor point.
- The **ground plate** has 1 port (**GP\_GND**). It looks like a rectangle overlapping the entire ground plate.
- The **bonding pad** has 2 ports (**P\_GND**, **P\_MTL**). **P\_GND** looks like a long rectangle (2 units thick) stretching across the left side of the pad. **P\_MTL** looks like a long rectangle stretching across the right side of the pad.

Begin by connecting the **PL\_Left** and **PL\_Right** ports of **PlateInst** to the **C\_Free** ports of **CombLeft** and **CombRight**.

- ☑ Choose the **Box** tool  and select the **Poly1** layer from the **Layer Palette** by clicking it. **Poly1** should be the first item in the fourth row from the top of the **Layer Palette**. As you cover **Poly1**, the tool tip will read the layer name. **Poly1** will also appear in the list box at the top of the **Layer Palette**.
- ☑ Draw a box covering the **PL\_Right** port of **PlateInst** and the **C\_Free** port of **CombRight**. Click once to set the lower left corner, hold the key down and drag to the opposite corner, and release.
- ☑ Draw a box covering the **PL\_Left** port of **PlateInst** and the **C\_Free** port of **CombLeft**.

We will now connect the **PL\_Top** and **PL\_Bottom** ports of **Platelnst** to the **FS\_Free** ports of **SpringTop** and **SpringBottom**.

- ☑ Connect the top folded spring to the plate by drawing a box on **Poly1** covering the **FS\_Free** port of **SpringTop** and the **PL\_Top** port of **Platelnst**.
- ☑ Connect the bottom folded spring to the plate by drawing a box on **Poly1** covering the **FS\_Free** port of **SpringBottom** and the **PL\_Bottom** port of **Platelnst**.

Finally, we will connect the **FS\_Free** ports of **SpringTop** and **SpringBottom** to the **GP\_GND** port of **GroundPlatelnst**.

- ☑ Connect the ports of the two folded springs to the ground plate by drawing a box on the **Poly0** layer covering the **GP\_GND** port of **GroundPlatelnst**. This box should cover all of **GroundPlatelnst**.

Now all the connections will be properly recognized by L-Edit/Extract.

## Extracting Layout

*Layout extraction* produces a SPICE netlist consisting of device and connectivity information used for comparing layout vs. schematic (LVS) or SPICE simulations. Design rule checking (DRC) ensures that a layout conforms to fabrication process requirements, but it does not verify that the layout actually implements what was intended; nor does it assist in determining whether the system will perform to your specifications.

Extracting MEMS designs involves the use of the MEMS Pro feature called *subcircuit extract*. Subcircuit extract involves the extraction of subcircuit cells as “black boxes” with connection ports and cell properties.

### Note

---

For more information see Subcircuit Recognition on page 3-73 and Extracting Layout on page 3-48 of the *L-Edit User Guide*.

---



Select **Tools > Extract** to invoke the **Extract** tabbed dialog box.

An **extract definition file** must be loaded to provide technology information about your design. It contains a list of the connections and devices to be extracted.



- ☑ Click the **Browse** button and use the windows browser to select **mumps.ext** from the **tutorial** directory and enter **Layout.spc** as the output name.

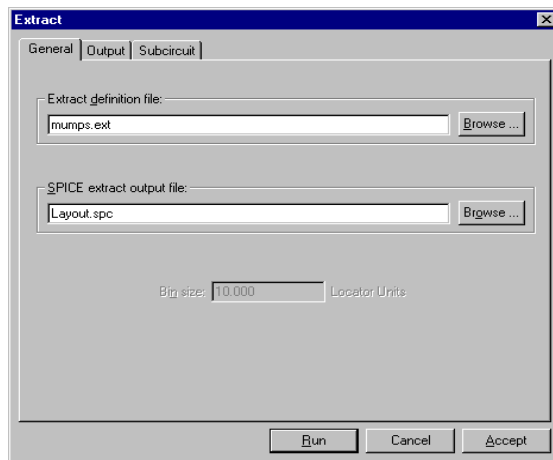


Figure 100: Selecting the extract definition file

- ☑ Click the **Subcircuit** tab.

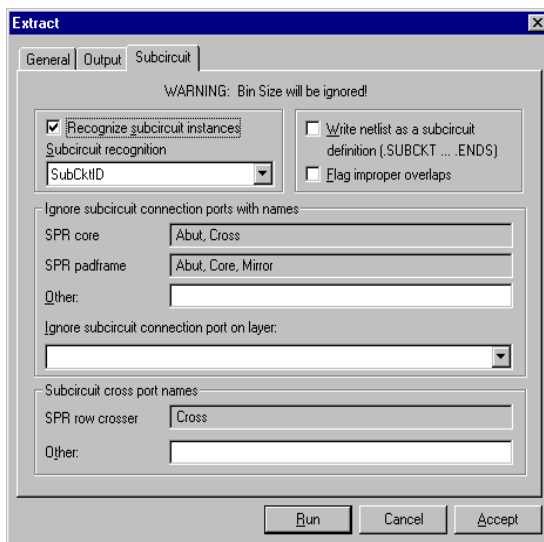


Figure 101: **Subcircuit** tab of the **Extract** dialog box

- ☑ Check the **Recognize Subcircuit Instances** checkbox.
- ☑ Select **SubCktID** as the **Subcircuit Recognition Layer**, uncheck the **Flag Improper Overlaps** checkbox.

- ✓ Click the **Run** button to begin the layout extraction.

Note that clicking **OK** will save the setup information but will not run the extraction. After clicking **Run**, a netlist file called **Layout.spc** will be created. This is a text file in SPICE format containing the devices extracted, their connectivity, and device geometrical parameter information. The netlist file can be used to run T-Spice simulations or to perform layout versus schematic verification (LVS).

- ☒ When the **L-Edit Warning** dialog appears, click the **Ignore All** button.

- ☒ Open the **Layout.spc** file selecting **File > Open**. In the **Open** dialog box, change the **File Type** to **Spice Files (\*.sp, \*.spc)**, select **Layout.spc** from the file list, and click **OK**. A text window is brought up in L-Edit containing the file. Examine the extracted file by using the scroll bar on the text window.

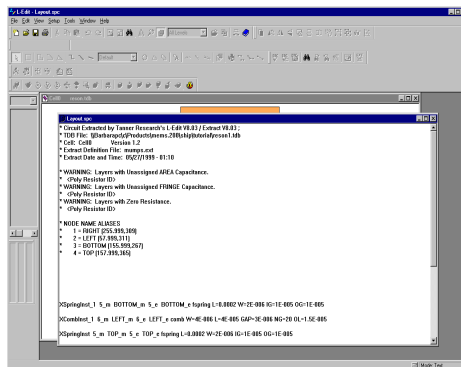


Figure 102: Viewing the extracted file

- ☒ Select **File > Exit** to exit L-Edit.

## Extracting Schematic for LVS

To export a schematic netlist for use in LVS, the schematic must contain only the device components and be free of all stimuli and simulation commands. You will now re-open the tutorial file.

- ☒ Open S-Edit.
- ☒ Select **File > Open** to open the **reson.sdb** file.

- ☒ Select **Module > Open** to open the **SchemLVS** module.

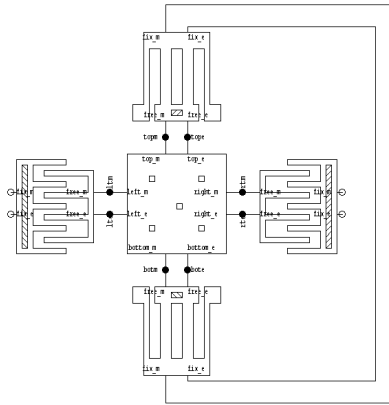


Figure 103: Schematic view of the resonator

- ☒ Select **File > Export** to invoke the **Export Netlist** dialog.

- ☑ In the **Export Netlist** dialog, choose **Pin number order** for the **Netlist Port Order** and uncheck the **Enable waveform probing** checkbox. Click **OK**.

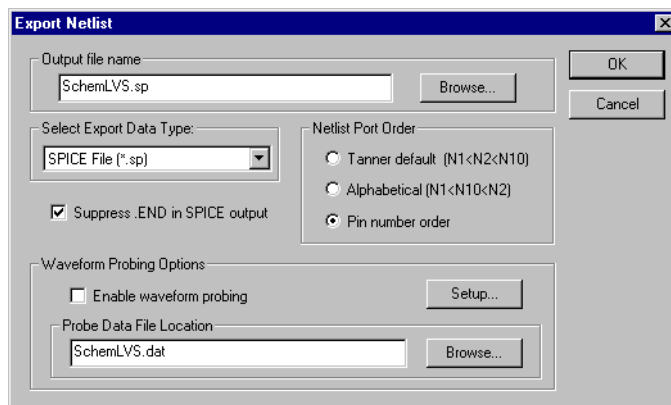



Figure 104: **Export Netlist** dialog

Clicking **OK** will save the setup information, run the extraction, and create an output netlist file that is preloaded with the module name. In this case, the file name will be **SchemLVS.sp**. This is a text file in SPICE format that contains device descriptions, their connectivity, and geometrical parameter information. The netlist file can be used to perform layout versus schematic verification (LVS).

- ☑ Select **File > Exit** to exit S-Edit.

## Comparing Netlists

An important step in the MEMS design process involves comparing the layout and the schematic to ensure that they describe the same system. This is *layout versus schematic comparison*, performed by comparing two *netlists* — one derived from the layout and one from the schematic.

- ☑ Double-click the **LVS** icon  to launch LVS.
- ☑ Select **File > Open** to open the **reson.vdb** file in the **tutorial** directory. The file contains predefined parameters to compare the SPICE files you created.

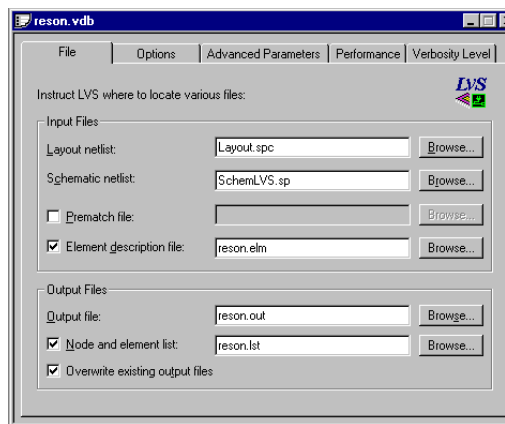





Figure 105: Viewing the preset parameters of the reson.vdb file

- ☑ Click the **Run** button  located on the toolbar to launch the comparison. The verification window will appear displaying the results.

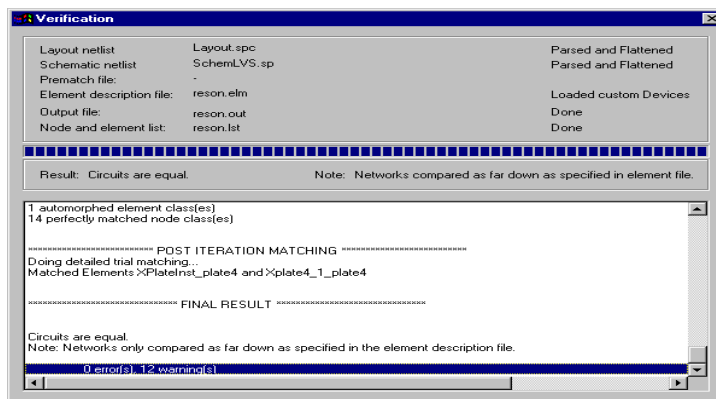


Figure 106: Verification window

The netlist generated from the schematic and the netlist generated from the layout are identical.

- ☑ Select **File > Exit** to exit LVS.

**Note**

---

For more information on *netlist comparison*, see Netlist Comparison on page 3-139 and Resolving Discrepancies on page 3-157 of the *L-Edit User Guide*.

---

# 12 Command Tool

---

- Introduction 475
- Accessing the Command Tool 477
- Command Tool Dialog 479

# Introduction

The **Command Tool** provides a graphical interface for entering device and model statements, simulation stimuli, commands, and options in SPICE syntax. The **Command Tool** is accessible from both T-Spice and S-Edit.

In S-Edit, the **Command Tool** forms the grammatically correct SPICE commands for use in Schematic mode, Symbol mode or in output property strings.

In T-Spice, the **Command Tool** can also be used to insert grammatically correct SPICE commands into your netlist. For more information on accessing the **Command Tool** in T-Spice see Simulation Commands on page 168 of the *T-Spice User Guide and Reference*.

## Usage in S-Edit

### *Schematic Mode*

In Schematic mode, the **Command Tool** is used to add SPICE commands which can be passed, via netlist export, to the T-Spice simulator. This simplifies the task of entering complex simulation commands such as those for parameter sweeps and optimization. It also allows the user to maintain, within the schematic database, symbols, schematics, and simulation parameters.

## *Symbol Mode*

In Symbol mode, the **Command Tool** can be used to associate a SPICE command with a symbol. This capability is of use to library designers who might want to associate an often used SPICE command with a symbol.

## *Property Creation*

The **Command Tool** can be used to set the value of an arbitrary property for use in a schematic symbol. With this use, the name of property is not limited to SPICE OUTPUT.

# Accessing the Command Tool

## Schematic Tools Toolbar

In the Schematic mode of S-Edit, the **Command Tool** is activated by clicking the **Command Tool** button on the **Schematic Tools** toolbar. Once the button is clicked, it remains depressed until another tool is activated by clicking any of the other buttons on the same toolbar.



Command Tool

When the **Command** tool is active, the mouse cursor becomes a cross-hair as it is dragged over the work space. A left-click on the work space will invoke the **T-Spice Command Tool** dialog.

## Module Menu

In the Schematic mode of S-Edit, the **Command Tool** may also be activated by selecting **Module > Command Tool**.

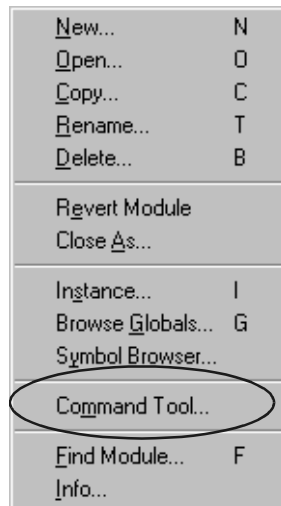


Figure 107: Selecting the **Command Tool** Option of the **Module** Menu

Selecting **Module > Command Tool** will invoke the **T-Spice Command Tool** dialog.

# Command Tool Dialog

The graphical interface for the **Command Tool** is the **T-Spice Command Tool** dialog (Figure 108).

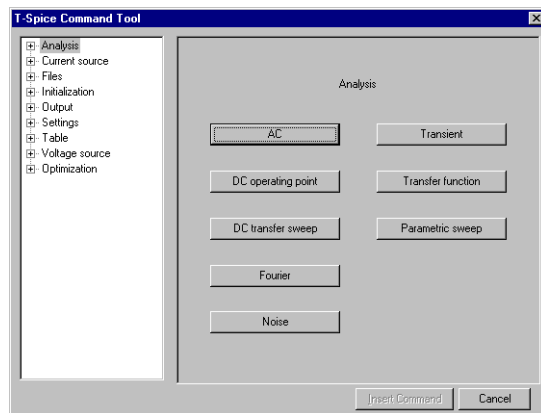


Figure 108: **T-Spice Command Tool** dialog

The left side of the dialog displays a hierarchical list of command categories. Double-clicking on a category (or clicking the plus or minus sign next to the category name) expands or collapses the list of specific commands. The right side of the dialog displays the list of commands corresponding to the selected



category. When a specific command is highlighted on the left side, or a command is selected from the command list on the right hand side, the right side of the dialog contains a field for each of the variables required by the command. The T-Spice command is generated from your dialog entry for the command variables.

The following example is a **T-Spice Command Tool** dialog for a transient analysis (Figure 109). This is done by clicking the plus sign in front of the **Analysis** category to expand the command list and selecting **Transient**. The command options dialog will replace the category list on the right side of the dialog. In the example, the transient analysis command options have been filled in.

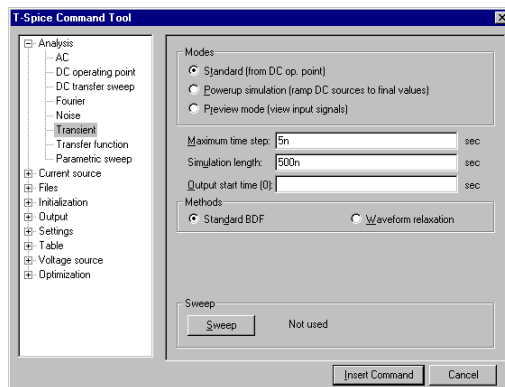


Figure 109: Customizing the transient analysis

Once the command parameters are set, clicking the **Insert Command** button will create a grammatically correct SPICE command string and a schematic object is created that contains the SPICE command string.



## Schematic Object Creation

As described in Command Tool Dialog on page 479, a SPICE command string will be formulated from your entries to the **T-Spice Command Tool** dialog box. In the Schematic mode, an instance of a template module, described in Template Module on page 482, will be created and placed at the cursor location if the **Command Tool** was accessed from the **Schematic Toolbar** or at the origin of the schematic window if the **Command Tool** was accessed from the **Module** menu. The command you specified using the **T-Spice Command Tool** dialog will be inserted as the **Value** of the **SPICE OUTPUT** property of the new instance.

### *Template Module*

The template module is provided as a part of the schematic library and is named **TSPICE COMMAND**. If this module does not exist within the current design space, a browse dialog will prompt you to enter the name of the module to use as the template. If the chosen template module does not contain a **SPICE OUTPUT** property, one will be created and placed at the origin of the symbol page with the **Value** set to “”.

## Symbol Mode

The method of access and the result of the **Command Tool** is similar for both Schematic and Symbol modes. The exception is that in the Schematic mode, an instance is created as the schematic object and in the Symbol mode, a property is created instead.

The **Command Tool** is activated by clicking the **Command Tool** button on the **Schematic Tools** toolbar as before.

The **Command Tool** may also be accessed by selecting **Module > Command Tool**.

## Schematic Object in Symbol Mode

As described in Command Tool Dialog on page 479, a SPICE command string will be formulated from your entries to the dialog box and a property named **SPICE OUTPUT** of type **Text** will be created and placed at the cursor location if the **Command Tool** was accessed from the **Schematic Toolbar** or at the origin if the **Command Tool** was accessed from the **Module** menu. The **Text Size** is set to the **Default Port Text Size** and the **Show Format** is set to **None**. The **Value** of this property is defined by your entry into the **T-Spice Command Tool** dialog. Any existing entry in the **Value** field will be overwritten by the formulated command string.

If the **SPICE OUTPUT** property already exists, an error message will be displayed and the operation will be terminated.

## Create Property Dialog

In the symbol view mode of S-Edit, you may also access the **Command Tool** while creating a new property. To create a new property, the **Properties** tool must be activated by clicking the **Properties** button on the **Schematic Tools** toolbar. Once the button is clicked, it remains depressed until another tool is activated by clicking any of the other buttons on the same toolbar. When the **Properties** tool is active, the mouse cursor becomes a cross-hair. Clicking somewhere in the work space will invoke the **Create Property** dialog.

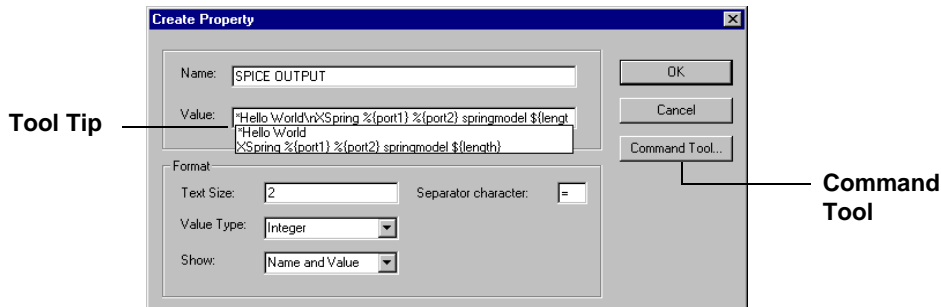


Figure 110: Create Property dialog

Pressing the **Command Tool** button in the **Create Property** dialog will invoke the **T-Spice Command Tool** dialog.

As described in Command Tool Dialog on page 479, a SPICE command string will be formulated from your entries to the dialog box and it will be inserted into the **Value** field, replacing the previous content. Clicking **OK** will create a property placed at the cursor location.

# 13      **Block Place and Route Tutorial**

---

- Initializing the Design 488
- Routing the Design 499

This tutorial demonstrates some key steps in the Block Place and Route (BPR) design flow for use with MEMS.

BPR assists in the design of systems by providing an automated means of placing and routing the blocks that compose them. Blocks may contain the layout of MEMS sensors, amplifiers, demodulators, oscillators, A-D converters, and so forth. BPR enables you to focus on the design issues of individual blocks, and when those designs are functional, helps you gather the pieces and connect them

in the configuration you desire. Some features of BPR, including timing analysis, signal integrity analysis, and layout verification will help you optimize your entire design to an overall system performance goal.

The BPR process consists of four stages: design preparation, initialization, placement, and routing. Optional steps include timing analysis, signal integrity analysis and layout verification.

Placing and Routing Block Designs on page 2-144 of the *L-Edit User Guide* gives a detailed description of the BPR design process and terminology, including a tutorial based on a CMOS adder circuit that demonstrates BPR initialization, routing, moving vias and routing wires, the netlist navigator, and assisted manual routing.

In this tutorial, you will be placing and routing the blocks that compose a Q-controlled resonator system. This system consists of a MEMS linear resonator and interface circuitry that includes several transistors, a resistor, and a capacitor.

Two example files are located at **<install directory>\Examples\Bpr**. The first file, **mems.tbd**, is a source for setup information as you create your own tutorial file. The second file, **mems\_placed.tbd**, is used to demonstrate automatic routing.



## Initializing the Design

During BPR initialization, L-Edit reads a netlist, places the cells in the design in a special top-level BPR cell, and displays their connectivity as routing guides.

You must first specify the netlist that L-Edit will read for connectivity information. All the cells appearing within the netlist must be present in your design file.

In this part of the tutorial you will:

- Use the design navigator to copy cells into a design
- Enter initialization values
- Set a top-level BPR cell



Launch L-Edit.

L-Edit opens with **Cell0** of a new empty file called **Layout1**. You will add cell information to **Layout1** as part of the initialization process.

- ☒ Select **File > Replace Setup**.

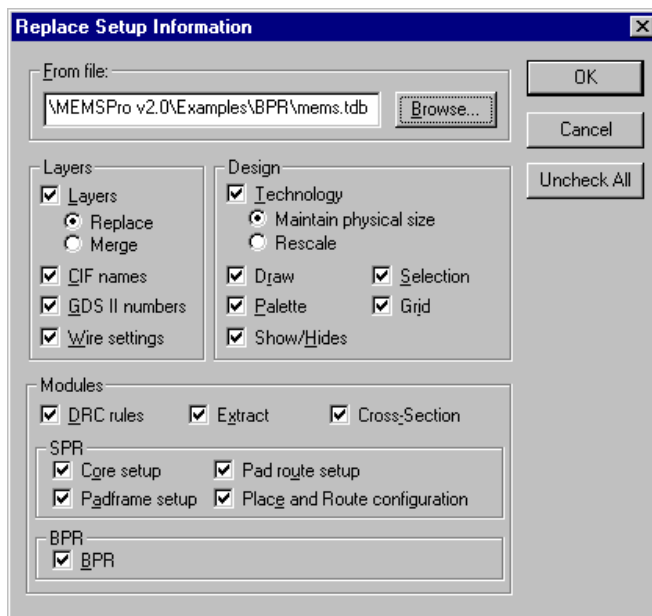


Figure 111: **Replace Setup Information** dialog

All the boxes to import values from your design should be checked, these describe the palette, application, design, and layer setup information that will be incorporated into the **Layout1** file.

- ☑ Browse to **<install directory>\Examples\Bpr** and select the file ***mems.tdb***.
- ☑ Click **OK** to close the **Replace Setup Information** dialog. The setup information has been transferred to your file.
- ☑ Use **File > Save** to save your file in the **<install directory>\Examples\Bpr** subdirectory as ***tutorial.tdb***.

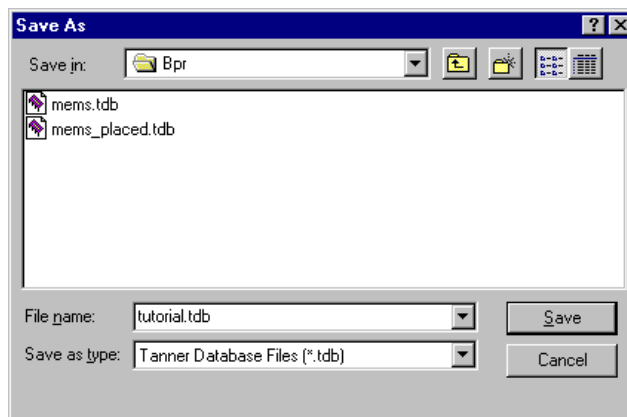


Figure 112: Saving the setup information

In order to initialize a design for BPR, all the cells referenced in the netlist must exist in the active layout file. You will copy the cells that you will need for

initialization into the ***tutorial.tdb*** file using L-Edit's **Design Navigator**. The **Design Navigator** lists all the cells included within a single design, and allows you to browse among them.

- ☒ Use **View > Design Navigator** to open the **Design Navigator** for ***tutorial.tdb***.

- ☑ Use **File > Open** to open the **mems.tdb** file in the **<install directory>\Examples\Bpr** subdirectory. The **Design Navigator** for **mems.tdb** should appear.

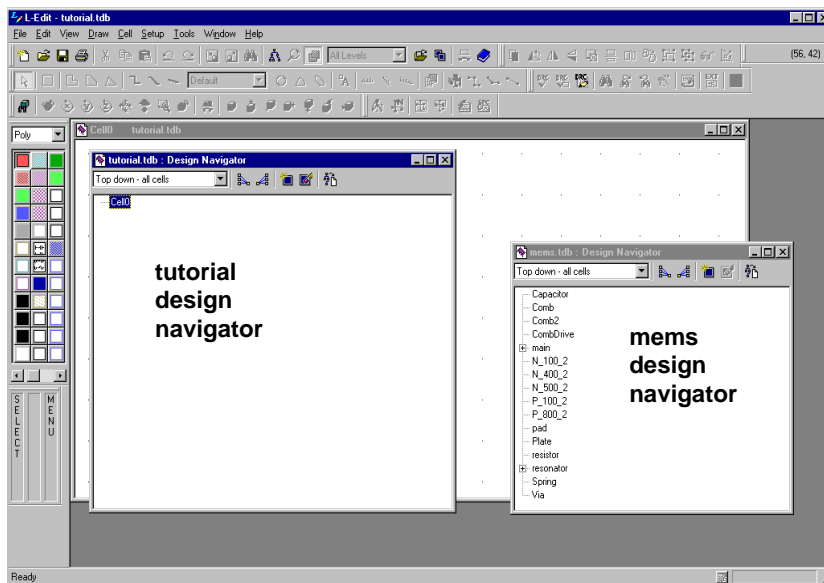


Figure 113: Tutorial design navigator and mems design navigator

- ☑ From the **mems.tdb** file's **Design Navigator**, select the **Capacitor** and drag and drop it into the **tutorial.tdb** file's **Design Navigator**.

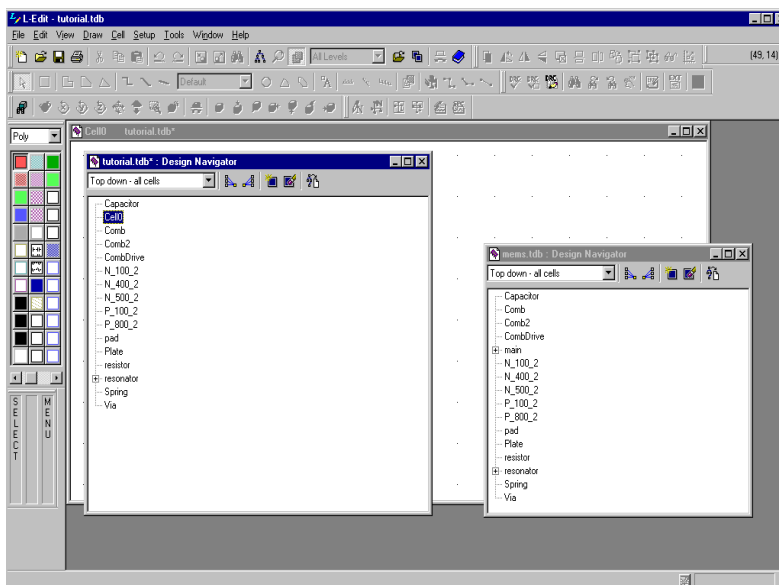


Figure 114: Copying cells from a database to another

- ☑ Likewise, drag and drop the cells **N\_100\_2**, **N\_400\_2**, **N\_500\_2**, **P\_100\_2**, **P\_800\_2**, **pad**, **resistor**, **resonator**, and **Via** into **tutorial.tdb**.

**Warning**

---

*Do not copy the **main** cell.*


---

**Note**

---

Since the **Spring** is referenced by the resonator, it is automatically copied.

---

- ☒ The cells you copied include a capacitor, three NMOS transistors, two PMOS transistors, a pad, a resistor, a resonator, and a via.
- ☒ Use **Ctrl+S** to save your **tutorial.tdb** file.
- ☒ Close both **Design Navigator** windows by clicking the  button on the upper right corner of the windows.

- ☑ Use **Cell > New** to create a cell for use during initialization. Enter ***top-level*** in the **New cell name** field.

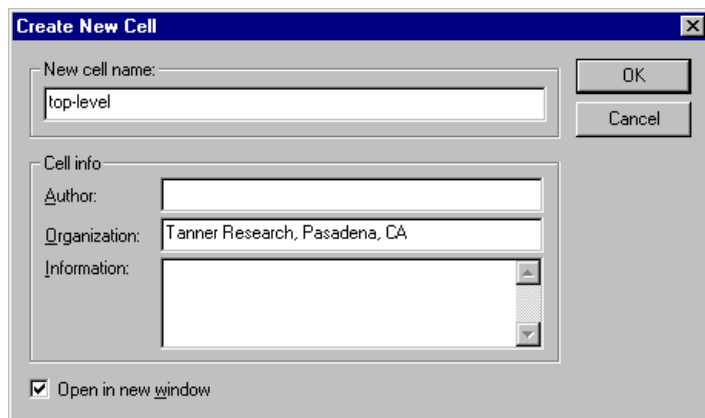


Figure 115: **Create New Cell** dialog

- ☑ Click **OK** to close the **Create New Cell** dialog. The cell ***top-level*** should now be the active cell.
- ☑ Use **Tools > BPR > Initialization** to open the **BPR Initialization** dialog.



In the **BPR Initialization** dialog, you will enter a netlist, assign the default signal type, specify a top-level I/O cell, set a routing pitch and pick a routing guide layer.

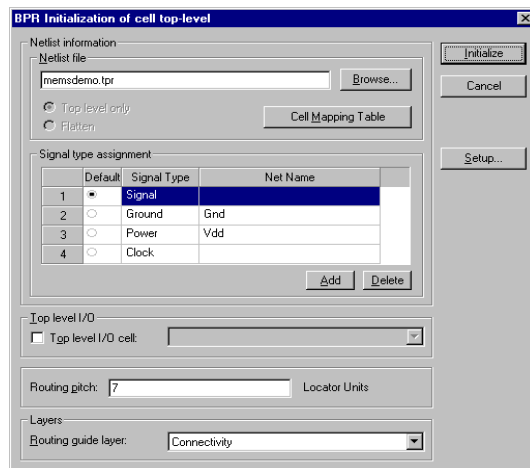


Figure 116: **BPR Initialization** dialog

- ☑ In the **Netlist file** group, browse to the <install directory>\Examples\Bpr subdirectory and select the **memdemo.tpr** netlist. Confirm that the **BPR Initialization of cell top-level** fields appear as shown above. Select **Signal** as the **Default Signal Type**. Select **Connectivity** from the layers in the **Routing guide layer** pull-down list. Enter a routing pitch of **7**. Click **Initialize**.

- ☑ Maximize the active window and press the **Home** key to center the image. The initialized design should look like the following:

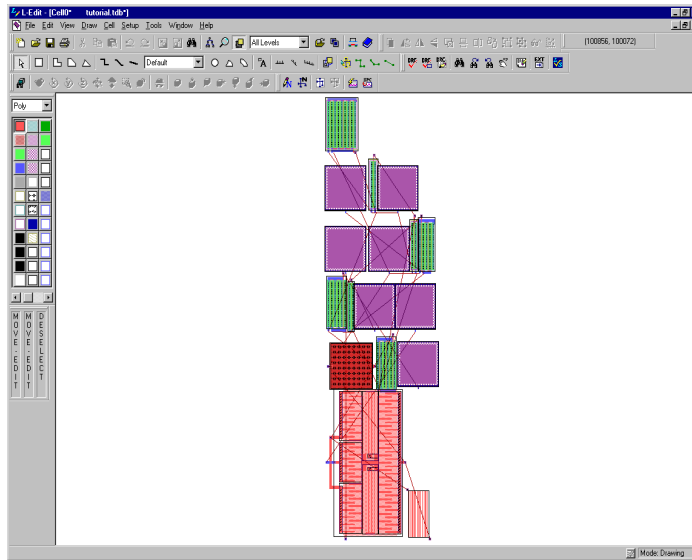


Figure 117: Layout view of the initialized design

- ☑ Save and close the **tutorial.tdb** file.

You've successfully initialized your BPR cell. BPR has placed the blocks listed in the netlist file. The connectivity is displayed as a network of routing guides for pin-to-pin connections, and will update interactively as you move the blocks.



## Routing the Design

After initialization, you can manually or automatically move and connect the blocks in your design to the positions you desire. In this tutorial, we demonstrate automatic routing of a placed file (***mems\_placed.tdb***).

The autorouter can route all the nets in a design or a restricted set of selected nets. Assisted manual routing tools are useful for hand-routing nets when you need more control over their exact placement; for example, when you want to reduce parasitic capacitances and resistances around a MEMS sensor and its interface circuitry.

In this portion of the tutorial, you will:

- Define routing layers and via cells, set wire widths, select keep-out and subcircuit recognition layers, set the routing pitch, and define any excluded signals
- Use the automatic router to route an entire design

- ☑ Use **File > Open** to open the **mems\_placed.tdb** file in the **<install directory>\Examples\Bpr** subdirectory. Maximize the window and press the **Home** key so the design fills the window as shown below.

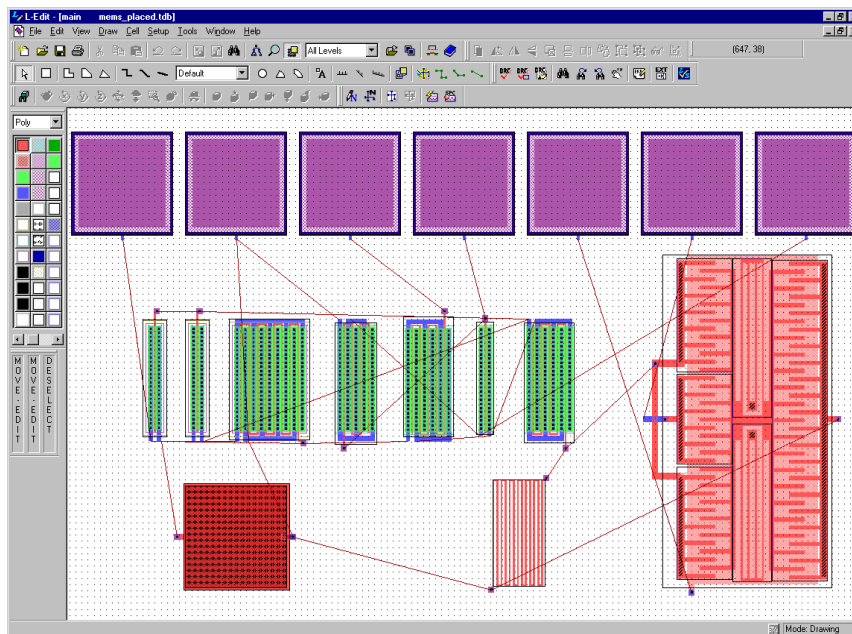


Figure 118: Layout view of the *mems\_placed.tdb* file

- ☑ Use **Tools > BPR > Setup** to confirm that the **General** tab fields are set as below. **Route selection type** should be set to **Net**, and the boxes should be checked.

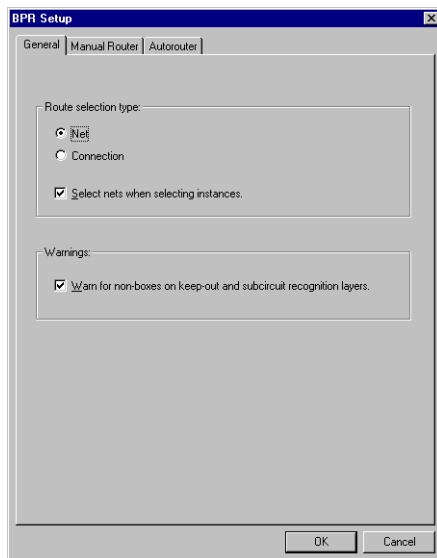


Figure 119: **General** tab of the **BPR Setup** dialog

- ☒ Select the **Autorouter** tab to confirm that fields in that tab of **BPR Setup** are set as shown below.

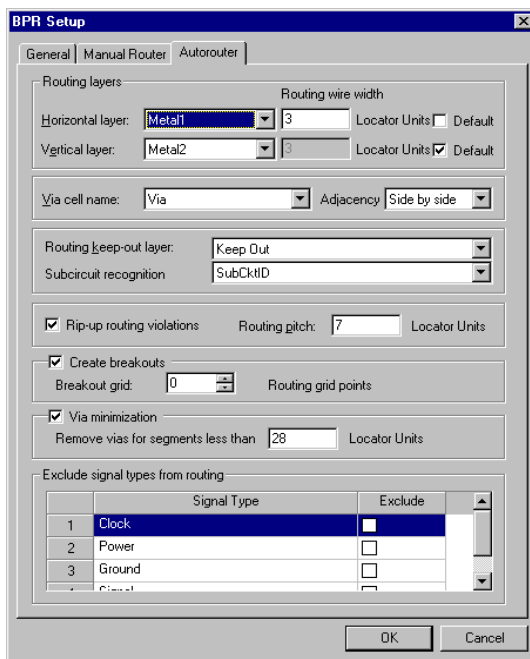


Figure 120: **Autorouter** tab of the **BPR Setup** dialog

- ☑ Click **OK** to close the **BPR Setup** dialog.
- ☑ Use **Tools > BPR > Route All** to automatically route all nets in the design.
- ☑ The following **BPR - Automatic Routing Report** appears when the router has completed its attempt. Note that 11 nets were completely routed, no nets were partially routed or not routed at all.

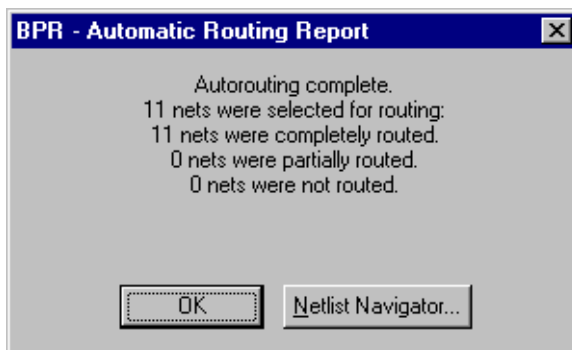


Figure 121: **BPR - Automatic Routing Report** dialog

- ☑ Click **OK** to close the routing report.



The routed design should look like the figure below. You have successfully completed the BPR tutorial.

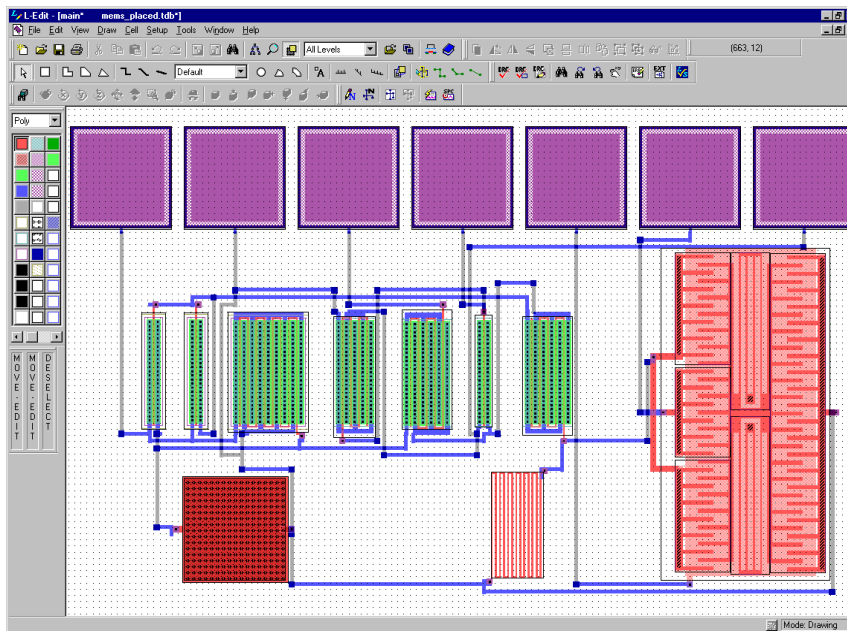


Figure 122: Layout view of the routed design

You may continue experimenting with BPR by removing the routed connections of the design (**Tools > BPR > Unroute All**), moving the blocks around, and then routing again.



# 14      Extending the MEMS Library

---

▪ Introduction	507
▪ Schematic Symbols	508
▪ SPICE Models	511
▪ Layout Generators	514

# Introduction

The MEMS library (MEMSLib) contains a variety of elements, also called primitives, that can be combined to create MEMS devices. These building blocks are listed in the chapter MEMSLib Reference on page 280 of the *MEMS Pro User Guide*. MEMSLib is continually updated to make available the widest possible selection of parts for generating MEMS devices. However, the possibilities of MEMS design are too broad for MEMSLib to completely represent all components of all possible devices. Our priority is to construct those components most often required for MEMS design.

A powerful feature of MEMS Pro is that our design library can be easily extended. We outline the process for adding new elements to the MEMS library in this chapter.

# Schematic Symbols


## Note

---

We frequently refer to S-Edit concepts, operations, and commands, all of which are more fully described in the *S-Edit User Guide and Reference*.

---

This section offers step-by-step instructions for creating the schematic symbol that you will use to reference the MEMS element you will design. To understand how you will produce the symbol, let's look at a symbol from our existing library.

- ☑ Double-click the **S-Edit** icon  to launch S-Edit.
- ☑ Select **File > Open**. Open the **reson.sdb** file in the tutorial directory (Figure 123).
- ☑ Select **Module > Open**. Select module **Plate4**.

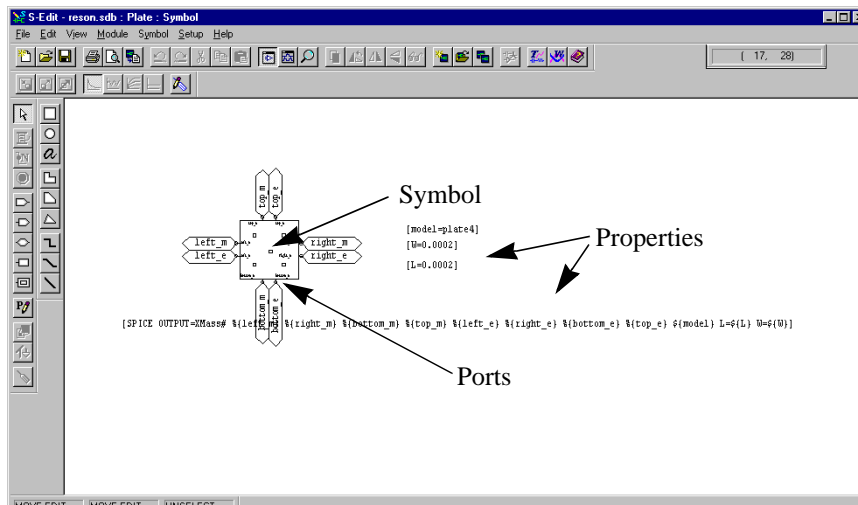


Figure 123: Symbolic view of the resonator

**Plate4** is composed of three parts: the symbol representing the element, its properties, and ports. The **SPICE OUTPUT** property (shown across the bottom of the S-Edit window) is essential for exporting to a SPICE netlist.

Perform the following steps to create a new symbol:

- ☑ Select **Module > New**.
- ☑ In the **Module Name** field, enter the name of your new symbol.
- ☑ Ensure that the current view is in symbol mode by selecting **View > Symbol Mode**.
- ☑ Draw your symbol with the **Annotation** tools.
- ☑ Place input and output ports on your symbols with the **Port** tool.
- ☑ Add properties to your symbol with the **Properties** tool.
- ☑ Switch to **Schematic** mode to complete the schematic for your symbol.

## SPICE Models



### Note

---

We frequently refer to T-Spice concepts, operations, and commands, all of which are more fully described in the *T-Spice User Guide and Reference*.

---

SPICE can, of course, be used to simulate electrical circuits. SPICE can also be used to simulate multiple domain systems if the non-electrical system can be characterized by differential equations. This section offers instructions for creating a behavioral model under a multiple domain system.

- 
- 
- ☑ Find the appropriate mapping between SPICE's electrical variables and the variables in the other domain. For example, to model behavior in the mechanical domain, force can be mapped to current and position to voltage.
  - ☑ Find the differential equation that best describes the element.
  - ☑ Create a subcircuit model that captures the differential equation. The subcircuit may contain external functional models and/or a network of electrical primitive components.
  - ☑ Test your model. Verify that you have captured the behavior as you intended.



## Application Example

A simple, one-dimensional linear spring may be modeled by a network of electrical primitives:

- ☑ Map force to current and position to voltage.
- ☑ The mechanical equation  $F = k \cdot x$  can be mapped to the electrical equation  $I = k \cdot V$  where the spring constant,  $k$ , is a function of the geometry of the spring. Electrically, the element is a resistor whose resistance is a function of the spring geometry.
- ☑ Mapped to the electrical domain,  $k$  becomes  $1/R$ ,  $I$  represents force  $F$ , and  $V$  represents position  $x$ . Therefore, the subcircuit model is:

```
.subckt LinearSpring n1e n2e n1m n2m k=1 Re=1k
Rmech n1m n2m '1/k'
Relect n1e n2e 'Re'
.ends
```

To use this model in a SPICE file, reference it by:

```
Xspring n1e n2e n1m n2m LinearSpring k=.5 Re=2k
```

The *external model* feature can model behavior that cannot be described in terms of electrical primitives, for instance, the behavior of a controlled source that depends on arbitrary functions of state variables.

**Note**

---

The *External Model* feature is fully described in the chapter entitled User-Defined External Model on page 634 of the *T-Spice User Guide and Reference*.

---

# Layout Generators

## Note

---

We frequently refer to L-Edit concepts, operations, and commands, all of which are more fully described in the *L-Edit User Guide*. For more detailed information on writing UPI code, see Programming the User Interface on page 4-10 of the *L-Edit User Guide*.

---

To learn how to create a new layout generator, follow the instructions in the L-Edit/UPI On-Line Tutorial. Here, we describe an existing layout generator.



## Sample Layout Generator



The following code generates a rectangular plate drawn on the **Poly1** layer from values supplied by the user. The plate is parameterized by its length and width. The macro is bound to the **F1** hot key. This source code can also be found in **<install directory>\Examples\lupi\plate.c**.

```
#include <stdlib.h>
#include <string.h>
#include "ldata.h"
#include "lupi_usr.h"

struct Plate_Struct {
```

```
        char name[20];
        char instname[20];
        int width;
        int length;
    };

    int  Get_Parameters_Plate ( struct Plate_Struct  *Structure
    );
    int  Create_Plate ( struct Plate_Struct plate );
    void Generate_Plate ( void );

    int  UPI_Entry_Point( void )
    {
        LMacro_BindToHotKey ( KEY_F1, "Generate, PLATE",
        "Generate_Plate" );
        return 1;
    }

    void Generate_Plate ( void )
    {
        struct Plate_Struct Plate;

        if ( !Get_Parameters_Plate ( &Plate ) ) return;
        if ( !Create_Plate ( Plate ) ) return;
        return;
    }

    int  Get_Parameters_Plate ( struct Plate_Struct  *Structure
    )
```

```

{
    LDialogItem Dialog_Items [ 3 ] =
        { { "name", "name" },
          { "width", "200" },
          { "length", "100 " } };
    if ( !LDialog_MultiLineInputBox ( "Plate Parameters",
    Dialog_Items, 3 ) )
        return 0;

    strcpy ( Structure->name, Dialog_Items[0].value );
    Structure->width = atoi ( Dialog_Items[1].value );
    Structure->length = atoi ( Dialog_Items[2].value );
    return 1;
}

int Create_Plate ( struct Plate_Struct plate )
{
    LCell Cell_Original = LCell_GetVisible ( );
    LFile File_Now = LCell_GetFile (
    Cell_Original );
    LLayer Layer_Poly1 = LLayer_Find ( File_Now,
    "Poly1" );
    LPoint Point_Cursor = LCursor_GetPosition ( );
    LCell Cell_Now;
    LTransform Plate_Xform;
    LInstance Plate_Inst;
    LMagnification NoMag;
    if ( LCell_Find(File_Now, plate.name) ) {

```

```
        LDialog_AlertBox( "Cell with that name exists!
EXITING!" );
        return 0;
    }

    Cell_Now = LCell_New( File_Now, plate.name );

    /* draw the plate */
    LBox_New ( Cell_Now, Layer_Poly1,
              0, 0, plate.length, plate.width);

    /* instance plate cell into current cell */
    NoMag.num = (LLen) 1;
    NoMag.denom = NoMag.num;
    Plate_Xform = LTransform_Set(Point_Cursor.x,
Point_Cursor.y, LNormalOrientation, NoMag);
    Plate_Inst = LInstance_New(Cell_Original, Cell_Now,
Plate_Xform, LPoint_Set(1,1) , LPoint_Set(0,0));
    LInstance_SetName( Cell_Original, Plate_Inst,
plate.instname );
    LCell_MakeVisible(Cell_Original);
    return 1;
}
```

# 15 MEMSLib Reference

---

▪ Introduction	519
▪ Using the MEMS Library	524
▪ Active Elements	532
▪ Passive Elements	560
▪ Test Elements	578
▪ Resonator Elements	601

# Introduction

MEMSLib provides a library of components from which full MEMS devices can be built. The library provides schematic symbols, export to SPICE capability, SPICE models and parameterized layout generators. The layout generation part of this library is largely based on creating a graphical user interface to the Consolidated Micromechanical Elements Library (CaMEL) developed at MCNC.

The layouts for the library elements are automatically generated based on some user input parameters using the MEMS parameterized layout generator macro. The layout generators assume a two-layer surface micromachined process with two structural layers, two sacrificial layers, and two electrical connect layers. The default technology setting is for MUMPS. Macro usage information can be found in MEMSLib Layout Macros on page 257. Schematic designs are created using MEMSLib by instantiating the MEMSLib symbol modules. Simulations results can be viewed directly from the schematic they model using the waveform probing feature. Further instruction on usage of the library elements appear in Using the MEMS Library on page 524 of this chapter.

This library reference provides descriptions, file locations, and parameter lists (including default parameter values) for each library element. The corresponding layout palette button, the L-Edit/UPI parameter input dialog box, and illustrated geometry for each element are also shown. The layout library example cell for each element was generated using default parameter values.



<i><b>Element</b></i>	<i><b>Description</b></i>
Library	Accessing the MEMS Library Palette on page 526
<b>Active Elements</b>	
S_LCOMB	Linear Electrostatic Comb Drive Elements (S_LCOMB_1, S_LCOMB_2) on page 532
S_LSDM	Linear Side Drive Elements (S_LSDM_1, S_LSDM_2) on page 535
S_RCOMBU	Unidirectional Rotary Comb Drive Elements - Type 1 (S_RCOMBU_1, S_RCOMBU_2) on page 538
S_RCOMBUA	Unidirectional Rotary Comb Drive Elements - Type 2 (S_RCOMBUA_1, S_RCOMBUA_2) on page 542
S_RCOMBD	Bidirectional Rotary Comb Drive Elements (S_RCOMBD_1, S_RCOMBD_2) on page 546
S_RCDM	Rotary Comb Drive Elements (S_RCDM_1, S_RCDM_2) on page 550
S_RSDM	Rotary Side Drive Elements (S_RSDM_1, S_RSDM_2) on page 554
S_HSDM	Harmonic Side Drive Elements (S_HSDM_1, S_HSDM_2) on page 557
<b>Passive Elements</b>	
S_JBEARG_1	Journal Bearing Elements 1 (S_JBEARG_1) on page 560
S_JBEARG_2	Journal Bearing Elements 2 (S_JBEARG_2) on page 563

<b><i>Element</i></b>	<b><i>Description</i></b>
S_LCLS	Linear Crab Leg Suspension Elements - Type 1 (S_LCLS_1, S_LCLS_2) on page 566
S_LCLSB	Linear Crab Leg Suspension Elements - Type 2 (S_LCLSB_1, S_LCLSB_2) on page 569
S_LFBS	Linear Folded Beam Suspension Elements (S_LFBS_1, S_LFBS_2) on page 572
S_SPIRAL	Dual Archimedean Spiral Spring Elements (S_SPIRAL_1, S_SPIRAL_2) on page 575
<b>Test Elements</b>	
S_APTEST	Area-Perimeter Dielectric Isolation Test Structure Element (S_APTEST_1) on page 578
S_COTEST_1	Crossover Test Structure Element - Type 1 (S_COTEST_1) on page 581
S_COTEST_2	Crossover Test Structure Element - Type 2 (S_COTEST_2) on page 584
S_EUBEAM	Euler Column (Doubly Supported Beam) Elements (S_EUBEAM_1, S_EUBEAM_2) on page 587
S_EUBEAMS	Array of Euler Column Elements (S_EUBEAMS_1, S_EUBEAMS_2) on page 590
S_GRING	Guckel Ring Test Structure Elements (S_GURING_1, S_GURING_2) on page 593
S_GRINGS	Array of Guckel Ring Elements (S_GURINGS_1) on page 596
S_PAD	Multilayer Pad Element (S_PAD_1) on page 599
<b>Resonator Elements</b>	
S_PLATE_1	Plate (S_PLATE_1) on page 601
S_LCOMB_1	Comb Drive (S_LCOMB_3) on page 604

<i>Element</i>	<i>Description</i>
S_LFBS_3	Folded Spring (S_LFBS_3) on page 607
S_GDPLATE_1	Ground Plate (S_GDPLATE_1 ) on page 610
S_PAD_2	Bonding Pad (S_PAD_2) on page 612

## Acknowledgment

The layout generation portion of the MEMSLib library is based on the Consolidated Micromechanical Element Library, CaMEL, developed at MCNC and funded by the Defence Advanced Projects Agency contract DABT 63-93-C-0051. The CaMEL software and associated manual, “CaMEL User’s Guide,” by Ramaswamy Mahadevan & Allen Cowen are Copyright ©1994, 1997 by MCNC. The CaMEL software and portions of the CaMEL manual are reproduced here and distributed with permission from MCNC. Please read the attached CaMEL license and copyright.

For more information on MCNC or CaMEL, please refer to the following URLs:

- <http://www.mcnc.org/>
- <http://mems.mcnc.org/>
- <http://mems.mcnc.org/camel.html>

Copyright © 1994, 1996 by MCNC. All rights reserved.

By using the software, you, the Licensee, indicate that you have read, understood, and will comply with the terms listed below.

Permission to use, copy, and modify for internal, noncommercial purposes is hereby granted. Any distribution of this program or any part thereof is strictly prohibited without the prior written consent of MCNC.

Title to copyright to this software and to any associated documentation shall at all times remain with MCNC and Licensee agrees to preserve the same. Licensee agrees not to make any copies, in whole or part, except for the Licensee's internal noncommercial use. Licensee also agrees to place this copyright notice on any such copies.

MCNC makes no representation or warranties, express or implied. By way of example, but not limitation, MCNC makes no representation or warranties of merchantability or fitness for any particular purpose or that the use of the licensed software components or documentation will not infringe any patents, copyrights, trademarks or other rights. MCNC shall not be held liable for any liability nor for any direct, indirect, or consequential damages with respect to any claim by Licensee or any third party on account of or arising from this Agreement or use of this software.

PostScript® is a registered trademark of Adobe Systems Inc. GDSII is a trademark of Calma, Valid, Cadence. SUN and SunOS are trademarks of Sun Microsystems, Inc. UNIX is a trademark of AT&T Bell Laboratories.

## Using the MEMS Library

New users of the MEMS Library should first run through the MEMS Pro Tutorial on page 14. Listed below are high-level descriptions of the steps a user should take to use the MEMS Library to create a MEMS design.

- ☑ Create an S-Edit schematic design using symbols from the MEMS symbol library.
- ☑ Customize the properties of the symbols to meet the design requirements.
- ☑ Add stimulus and simulation conditions to the schematic.
- ☑ Import the process parameters for MUMPS by adding a “.include process.sp” statement to the schematic. Examining the test schematics in the **memslib.sdb** file may help illustrate the setting up of simulations using S-Edit.
- ☑ Export to SPICE netlist and run a T-Spice simulation.
- ☑ If the simulation results do not match the design requirements, iterate the process of modifying symbol properties and running simulations until the two match.
- ☑ Generate layout using the *MEMS Layout Palette*.
- ☑ Verify design.

The simulation model for each library element is documented in the schematic view of the symbol. The documentation of these models should be examined before use in order to fully understand the behavior of the element and to check for usage information. Note that some schematic elements (such as the side drive motor) must be built from composable elements. To simulate these elements, you may have to specify some external parameters.

**Note**

---

Review the release notes and the MEMS Application Notes for more information on the library elements and their use.

---

## Accessing the MEMS Library Palette

To access the MEMS Library Palette, select **Tools > Library Palette** in MEMS Pro Palette. The **Library Palette** dialog box should appear (Figure 124).

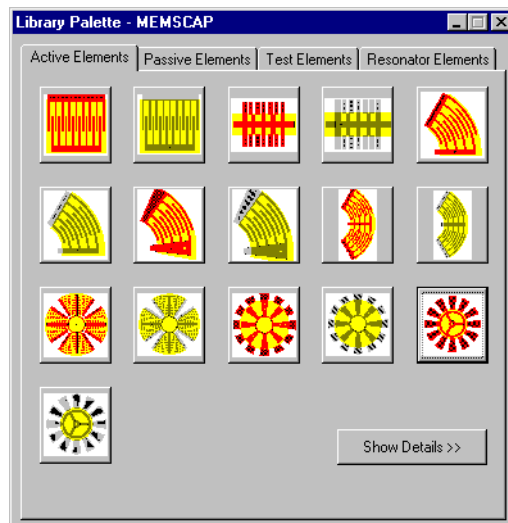
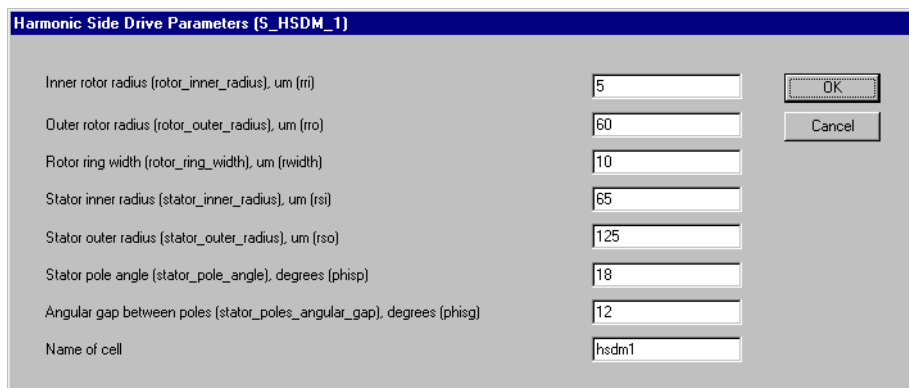


Figure 124: Library Palette

The Library Palette is a dialog box. Each tab corresponds to a set of MEMS elements. Categories include active, passive, and test elements, and resonators.

The layout generator for a particular element is executed by clicking the palette button corresponding to the desired element in the library palette. A parameters dialog box appears in which you can modify the parameters of the element. For instance, if you want to create a Harmonic Side Drive Motor, click the corresponding button and the following dialog box appears:



The dialog box is titled "Harmonic Side Drive Parameters [S\_HSDM\_1]". It contains seven parameter rows, each with a label, a text input field, and a unit. The input fields contain the following values: 5, 60, 10, 65, 125, 18, and 12. At the bottom right, there are "OK" and "Cancel" buttons.

Parameter	Value	Unit
Inner rotor radius (rotor_inner_radius)	5	um (ri)
Outer rotor radius (rotor_outer_radius)	60	um (rro)
Rotor ring width (rotor_ring_width)	10	um (rwidth)
Stator inner radius (stator_inner_radius)	65	um (rsi)
Stator outer radius (stator_outer_radius)	125	um (rso)
Stator pole angle (stator_pole_angle)	18	degrees (phisip)
Angular gap between poles (stator_poles_angular_gap)	12	degrees (phisig)
Name of cell	hsdm1	

Figure 125: **Harmonic Side Drive Parameters** dialog box



Once the parameters have been set, click **OK** to create the device layout in a new cell and instantiate it in the current cell.

## Show Details Button

The **Library Palette** dialog box contains a new button (the **Show Details** button). This button allows you to enlarge the **Library Palette** dialog box and displays the layout illustration for the selected element. If, for instance, you wish to view the layout illustration of the **Harmonic Side Drive Motor** (an active element), click the **Show Details** button and the **Library Palette** will be expanded (Figure 126).

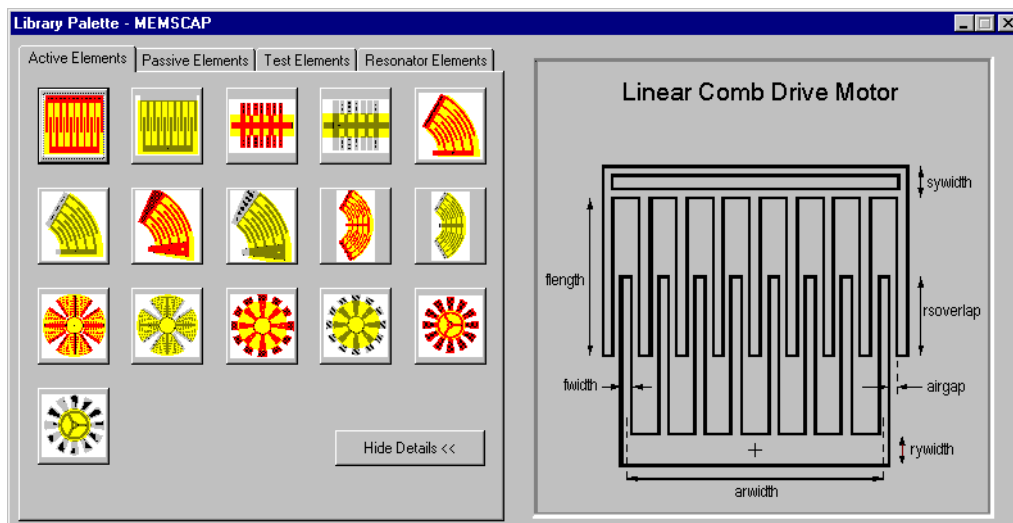
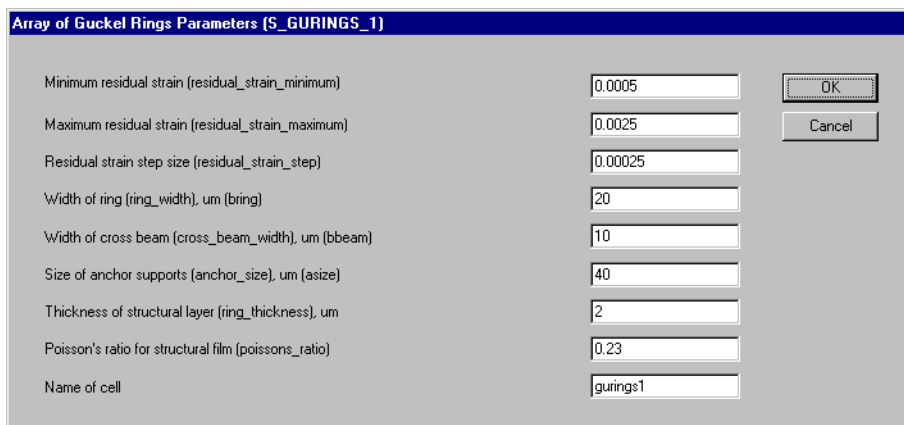


Figure 126: Enlarged **Library Palette** dialog box

The **Show Details** button has changed to the **Hide Details** button. By selecting this button, you revert the **Library Palette** dialog box to its initial size and the **Show Details** button appears again.

## Editing the Generated Layout Parameters

To edit a generated layout, select it and choose **Tools > Edit Component** in the MEMS Pro Palette. The device parameters dialog box (Figure 127) appears with the parameters values filled in. Change these values and click **OK**. The layout is automatically updated.



Array of Guckel Rings Parameters (\$_GURINGS_1)	
Minimum residual strain (residual_strain_minimum)	0.0005
Maximum residual strain (residual_strain_maximum)	0.0025
Residual strain step size (residual_strain_step)	0.00025
Width of ring (ring_width), um (bring)	20
Width of cross beam (cross_beam_width), um (bbeam)	10
Size of anchor supports (anchor_size), um (asize)	40
Thickness of structural layer (ring_thickness), um	2
Poisson's ratio for structural film (poissons_ratio)	0.23
Name of cell	guring1

Figure 127: Parameters dialog box for the array of Guckel rings

## MEMSLib L-Edit Library Page (Library)

*L-Edit: File **MEMSLIB.TDB** / Cell **Library** / Macro **MEMSLIB.DLL***

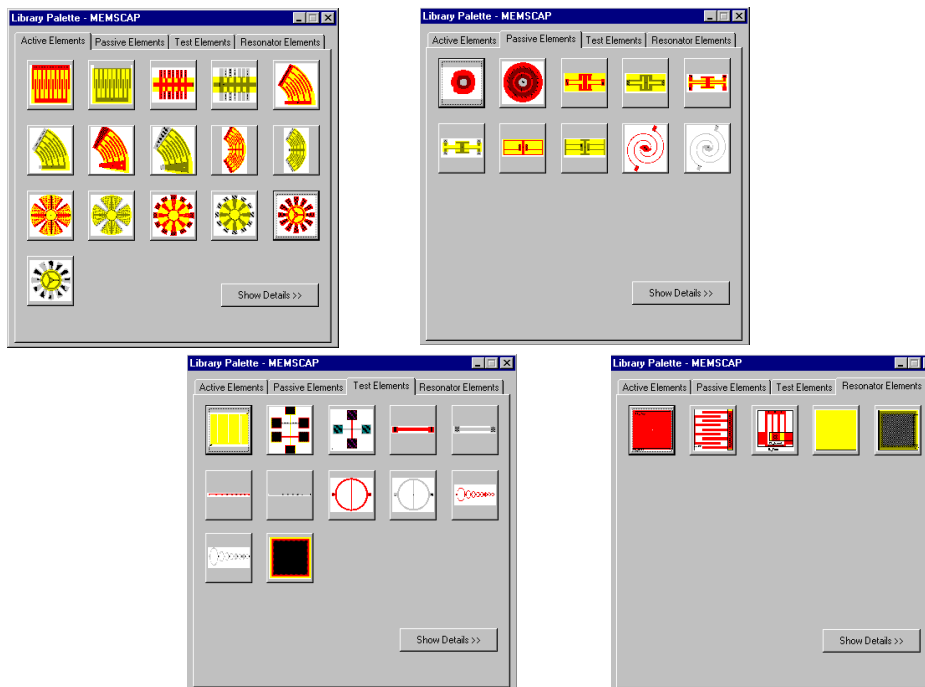


Figure 128: Various available library elements

## Active Elements

### Linear Electrostatic Comb Drive Elements (S\_LCOMB\_1, S\_LCOMB\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

#### Description

Generates a linear comb drive on the first (poly1) or second (poly2) structural layer.

The linear electrostatic comb drive that is designed on the first structural layer (S\_LCOMB\_1) has a corresponding schematic named S\_LCOMB\_1\_M\_X. The linear electrostatic comb drive designed on the second structural layer (S\_LCOMB\_2) has a corresponding schematic named S\_LCOMB\_2\_M\_X.

#### Parameter List

The following table provides the electrostatic comb drive parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Active rotor comb width	arwidth	98 $\mu\text{m}$	
Rotor yoke width	rywidth	12 $\mu\text{m}$	
Stator yoke width	sywidth	14 $\mu\text{m}$	
Length of comb fingers	flength	60 $\mu\text{m}$	finger_length
Width of comb fingers	fwidth	4 $\mu\text{m}$	finger_width
Air gap between fingers	airgap	3 $\mu\text{m}$	finger_gap
Stator-rotor finger overlap	rsoverlap	30 $\mu\text{m}$	finger_overlap
Direction of comb		1	DIR

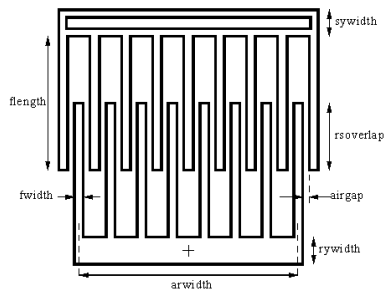
**S-Edit Symbol Name**      S\_LCOMB\_1\_M\_X (for poly1 layer), S\_LCOMB\_2\_M\_X (for poly2 layer)

## Linear Electrostatic Comb Drive Elements

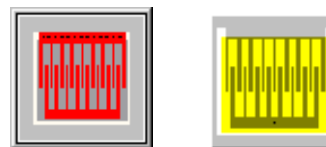
**Linear Comb Parameters (S\_LCOMB\_1)**

Active rotor comb width, um (arwidth)	<input type="text" value="98"/>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>
Rotor yoke width, um (rywidth)	<input type="text" value="12"/>	
Stator yoke width, um (sywidth)	<input type="text" value="14"/>	
Length of comb fingers (finger_length), um (flength)	<input type="text" value="60"/>	
Width of comb fingers (finger_width), um (fwidth)	<input type="text" value="4"/>	
Air gap between fingers (finger_gap), um (airgap)	<input type="text" value="3"/>	
Stator-rotor finger overlap (finger_overlap), um (rsoverlap)	<input type="text" value="30"/>	
Name of cell	<input type="text" value="lcomb1"/>	

Layout Parameter Entry Dialog Box



Layout Parameter Illustration



Layout Palette Buttons

## Linear Side Drive Elements (S\_LSDM\_1, S\_LSDM\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a linear side drive on the first or second structural layer.

### Parameter List

The following table provides the linear side drive parameters, their values and descriptions.

The linear side drive designed on the first structural layer (S\_LSDM\_1) has a corresponding schematic named S\_LSDM\_1\_M\_PHI. The linear side drive designed on the second structural layer (S\_LSDM\_2) has a corresponding schematic named S\_LSDM\_2\_M\_PHI..

Description	Layout Parameter Name	Default Value	Schematic parameter name
Active length of motor	mlength	120 $\mu\text{m}$	motor_active_length
Stator electrode width	swidth	12 $\mu\text{m}$	stator_pole_width



Description	Layout Parameter Name	Default Value	Schematic parameter name
Stator electrode pitch	spitch	20 $\mu\text{m}$	stator_pole_pitch
Stator electrode length	slength	40 $\mu\text{m}$	stator_pole_length
Rotor tooth width	rwidth	12 $\mu\text{m}$	rotor_pole_width
Rotor tooth pitch	rpitch	30 $\mu\text{m}$	rotor_pole_pitch
Rotor tooth height	rheight	20 $\mu\text{m}$	rotor_pole_height
Rotor yoke width	yokewidth	20 $\mu\text{m}$	
Air gap between stator and rotor	airgap	2 $\mu\text{m}$	rotor_stator_gap
Rotor offset with respect to stator	roffset	0 $\mu\text{m}$	
Number of gaps		3	number_of_gaps

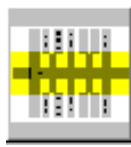
**S-Edit Symbol Name**     S\_LSDM\_1\_M\_X (for poly1 layer), S\_LSDM\_2\_M\_X (for poly2 layer)

## Linear Side Drive Elements

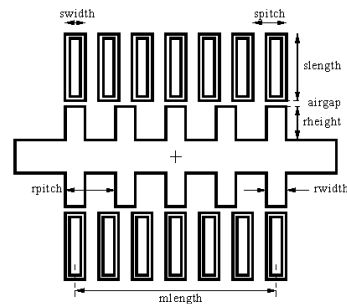
**Linear Side Drive Parameters (S\_LSDM\_1)**

Active length of motor (motor_active_length), um (mlength)	120	OK
Stator electrode width (stator_pole_width), um (swidth)	12	Cancel
Stator electrode pitch (stator_pole_pitch), um (spitch)	20	
Stator electrode length (stator_pole_length), um (slength)	40	
Rotor tooth width (rotor_pole_width), um (rwidth)	12	
Rotor tooth pitch (rotor_pole_pitch), um (rpitch)	30	
Rotor tooth height (rotor_pole_height), um (rheight)	20	
Rotor yoke width (rotor_yoke_width), um	20	
Air gap between stator and rotor (rotor_stator_gap), um (airgap)	2	
Rotor offset with respect to stator (rotor_stator_offset), um	0	
Name of cell	lsdm1	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Unidirectional Rotary Comb Drive Elements - Type 1 (S\_RCOMBU\_1, S\_RCOMBU\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a unidirectional rotary comb drive on the first or second structural layer.

The unidirectional rotary comb drive of type 1 designed on the first structural layer (S\_RCOMBU\_1) has a corresponding symbol (S\_RCOMBU\_1\_M\_PHI\_S) and a corresponding behavioral model (S\_RCOMBU\_1\_M\_PHI\_B). The unidirectional rotary comb drive of type 1 designed on the second structural layer (S\_RCOMBU\_2) has a corresponding schematic (S\_RCOMBU\_2\_M\_PHI\_S) and a corresponding behavioral model (S\_RCOMBU\_2\_M\_PHI\_B).

## Parameter List

The following table provides the unidirectional rotary comb drive (type 1) parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Active angular comb length	aclength	60 degrees	active_angular_length
Inner radius of rotor	rri	50 $\mu\text{m}$	rotor_inner_radius
Inner radius of stator comb	rsi	60 $\mu\text{m}$	stator_inner_radius
Outer radius of stator comb	rso	150 $\mu\text{m}$	stator_outer_radius
Rotor spoke width	rspokew	12 $\mu\text{m}$	rotor_spoke_width
Stator spoke width	sspokew	15 $\mu\text{m}$	stator_spoke_width
Width of comb fingers	fwidth	5 $\mu\text{m}$	finger_width
Air gap between adjacent comb fingers	airgap	5 $\mu\text{m}$	finger_gap
Angular finger overlap	trsovlp	30 degrees	finger_overlap
Direction of comb		1	DIR

**S-Edit Symbol Name**      S\_RCOMBU\_1\_M\_PHI\_S and S\_RCOMBU\_1\_M\_PHI\_B (for poly1 layer), S\_RCOMBU\_2\_M\_PHI\_S and S\_RCOMBU\_2\_M\_PHI\_B (for poly2 layer)

## Unidirectional Rotary Comb Drive Elements-Type1

**Unidirectional Rotary Comb Parameters (S\_RCOMBU\_1)**

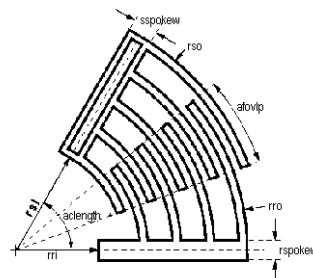
Active angular comb length (active_angular_length), um (aclength)	<input type="text" value="60"/>	<input type="button" value="OK"/>
Inner radius of rotor (rotor_inner_radius), um (ri)	<input type="text" value="50"/>	<input type="button" value="Cancel"/>
Inner radius of stator comb (stator_inner_radius), um (rsi)	<input type="text" value="60"/>	
Outer radius of stator comb (stator_outer_radius), um (rso)	<input type="text" value="150"/>	
Rotor spoke width (rotor_spoke_width), um (rspokew)	<input type="text" value="12"/>	
Stator spoke width (stator_spoke_width), um (spokew)	<input type="text" value="15"/>	
Width of comb fingers (finger_width), um	<input type="text" value="5"/>	
Asigap between adj. comb fingers (finger_gap), um	<input type="text" value="5"/>	
Angular finger overlap (finger_overlap), degrees (alovlp)	<input type="text" value="30"/>	
Name of cell	<input type="text" value="rcombu1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button

Unidirectional Rotary Comb Drive Motor



Layout Parameter Illustration

## Unidirectional Rotary Comb Drive Elements - Type 2 (S\_RCOMBUA\_1, S\_RCOMBUA\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a unidirectional rotary comb drive on the first or second structural layer. This element is similar to rcombu. The difference is in the design of the spoke. The center of the circular fingers is at (X-center, Y-center) and the rotor spoke is aligned with the X-axis.

The unidirectional rotary comb drive of type 2 designed on the first structural layer (S\_RCOMBUA\_1) has a corresponding symbol (S\_RCOMBUA\_1\_M\_PHI\_S) and a corresponding behavioral model (S\_RCOMBUA\_1\_M\_PHI\_B). The unidirectional rotary comb drive of type 2 designed on the second structural layer (S\_RCOMBUA\_2) has a corresponding schematic (S\_RCOMBUA\_2\_M\_PHI\_S) and a corresponding behavioral model (S\_RCOMBUA\_2\_M\_PHI\_B).

## Parameter List

The following table provides the unidirectional rotary side drive (type 2) parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
X location of center of comb fingers	xcenter	0	center_x_location
Y location of center of comb fingers	ycenter	0	center_y_location
Active angular comb length	aclength	60 degrees	active_angular_length
Inner radius of rotor	rri	50 $\mu\text{m}$	rotor_inner_radius
Inner radius of stator comb	rsi	60 $\mu\text{m}$	stator_inner_radius
Outer radius of stator comb	rso	150 $\mu\text{m}$	stator_outer_radius
Rotor spoke width	rspokew	12 $\mu\text{m}$	rotor_spoke_width
Stator spoke width	sspokew	15 $\mu\text{m}$	stator_spoke_width
Width of comb fingers	fwidht	5 $\mu\text{m}$	finger_width



Description	Layout Parameter Name	Default Value	Schematic parameter name
Air gap between adjacent comb fingers	airgap	5 $\mu\text{m}$	finger_gap
Angular finger overlap	trsfovlp	30 degrees	finger_overlap
Direction of comb drive		1	DIR

**S-Edit Symbol Name**     S\_RCOMBUA\_1\_M\_PHI\_S and S\_RCOMBUA\_1\_M\_PHI\_B (for poly1 layer), S\_RCOMBUA\_2\_M\_PHI\_S and S\_RCOMBUA\_2\_M\_PHI\_B (for poly2 layer)

## Unidirectional Rotary Comb Drive Elements - Type2

**Unidirectional Rotary Comb Parameters (S\_RCOMBUA\_1)**

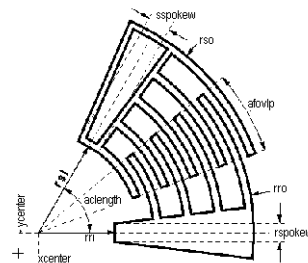
X location of comb fingers (center_x_location), um (xcenter)	0	OK
Y location of comb fingers (center_y_location), um (ycenter)	0	Cancel
Active angular comb length (active_angular_length), um (aclength)	60	
Inner radius of rotor (rotor_inner_radius), um (ri)	50	
Inner radius of stator comb (stator_inner_radius), um (rsi)	60	
Outer radius of stator comb (stator_outer_radius), um (rso)	150	
Rotor spoke width (rotor_spoke_width), um (rspokew)	12	
Stator spoke width (stator_spoke_width), um (sspokew)	15	
Width of comb fingers (finger_width), um	5	
Airgap between adj. comb fingers (finger_gap), um	5	
Angular finger overlap (finger_overlap), degrees (alovip)	30	
Name of cell	rcombuat	

Layout Parameter Entry Dialog Box

Unidirectional Rotary Comb Drive  
Motor



Layout Palette Button



Layout Parameter Illustration

## Bidirectional Rotary Comb Drive Elements (S\_RCOMBD\_1, S\_RCOMBD\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a bidirectional rotary comb drive on the first or second structural layer.

The bidirectional rotary comb drive designed on the first structural layer (S\_RCOMBD\_1) has a corresponding symbol (S\_RCOMBD\_1\_M\_PHI\_S) and a corresponding behavioral model (S\_RCOMBD\_1\_M\_PHI\_B). The bidirectional rotary comb drive designed on the second structural layer (S\_RCOMBD\_2) has a corresponding schematic (S\_RCOMBD\_2\_M\_PHI\_S) and a corresponding behavioral model (S\_RCOMBD\_2\_M\_PHI\_B).

### Parameter List

The following table provides the bidirectional rotary comb drive parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Active angular comb length	aclength	120 degrees	active_comb_length
Inner radius of rotor	rri	50 $\mu\text{m}$	rotor_inner_radius
Inner radius of stator comb	rsi	60 $\mu\text{m}$	stator_inner_radius
Outer radius of stator comb	rso	150 $\mu\text{m}$	stator_outer_radius
Rotor spoke width	rspokew	12 $\mu\text{m}$	rotor_spoke_width
Stator spoke width	sspokew	15 $\mu\text{m}$	stator_spoke_width
Width of comb fingers	fwidth	5 $\mu\text{m}$	finger_width
Airgap between adjacent comb fingers	airgap	5 $\mu\text{m}$	finger_gap
Angular finger overlap	afovlp	30 degrees	finger_overlap
Direction of combdrive		1	DIR

**S-Edit Symbol Name**     S\_RCOMBD\_1\_M\_PHI\_S and S\_RCOMBD\_1\_M\_PHI\_B (for poly1

layers), S\_RCOMBD\_2\_M\_PHI\_S and S\_RCOMBD\_2\_M\_PHI\_B (for poly2 layers)



## Bidirectional Rotary Comb Drive Elements

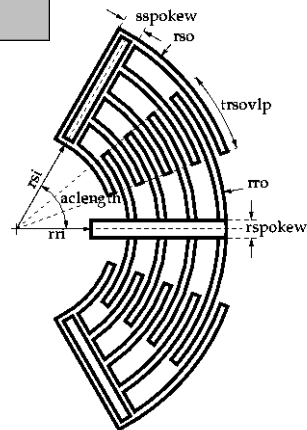
**Bidirectional Rotary Comb Parameters (S\_RCOMBD\_1)**

Active angular comb length (active_comb_length), degrees (aclength)	120	OK
Inner radius of rotor (rotor_inner_radius), um (ri)	50	Cancel
Inner radius of stator comb (stator_inner_radius), um (rsi)	60	
Outer radius of stator comb (stator_outer_radius), um (rso)	150	
Rotor spoke width (rotor_spoke_width), um (rspokew)	12	
Stator spoke width (stator_spoke_width), um (sspokew)	15	
Width of comb fingers (finger_width), um	5	
Airgap between adj. comb fingers (finger_gap), um	5	
Angular finger overlap (finger_overlap), degrees (trsovlp)	30	
Name of Cell	rcombd1	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Rotary Comb Drive Elements (S\_RCDM\_1, S\_RCDM\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a rotary comb drive on the first or second structural layer.

The rotary comb drive designed on the first structural layer (S\_RCDM\_1) has a corresponding symbol (S\_RCDM\_1\_M\_PHI\_S) and a corresponding behavioral model (S\_RCDM\_1\_M\_PHI\_B). The rotary comb drive designed on the second structural layer (S\_RCDM\_2) has a corresponding schematic (S\_RCDM\_2\_M\_PHI\_S) and a corresponding behavioral model (S\_RCDM\_2\_M\_PHI\_B).

### Parameter List

The following table provides the rotary comb drive parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Inner radius of rotor ring	rringi	38 $\mu\text{m}$	rotor_inner_radius
Outer radius of rotor ring	rringo	44 $\mu\text{m}$	rotor_outer_radius
Inner radius of stator comb	rsi	50 $\mu\text{m}$	stator_inner_radius
Outer radius of stator comb	rso	150 $\mu\text{m}$	stator_outer_radius
Width of comb fingers	fwidth	3 $\mu\text{m}$	finger_width
Airgap between adjacent comb fingers	airgap	3 $\mu\text{m}$	finger_gap
Rotor spoke width	rspokew	12 $\mu\text{m}$	rotor_spoke_width
Stator spoke width	sspokew	14 $\mu\text{m}$	stator_spoke_width
Gap between stator spokes at radius	sspokeg	5 $\mu\text{m}$	stator_spoke_gap
Stator overlap as a fraction of length	rsovlp	0.3	finger_overlap



<b>S-Edit Symbol Name</b>	S_RCDM_1_M_PHI_S and S_RCDM_1_M_PHI_B (for poly1 layer), S_RCDM_2_M_PHI_S and S_RCDM_2_M_PHI_B (for poly2 layer)
---------------------------	---

## Rotary Comb Drive Elements

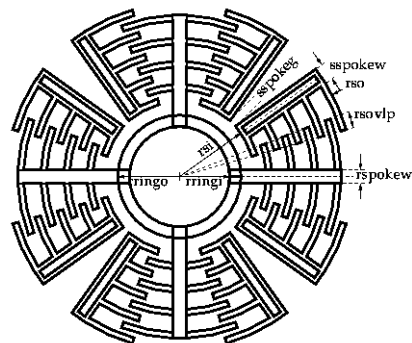
**Rotary Comb Drive Parameters S\_RCDM\_1**

Inner radius of rotor ring (rotor_inner_radius), um (ringi)	38	OK Cancel
Outer radius of rotor ring (rotor_outer_radius), um (ringo)	44	
Inner radius of stator comb (stator_inner_radius), um (rsi)	50	
Outer radius of stator comb (stator_outer_radius), um (rso)	150	
Width of comb fingers (finger_width), um	3	
Airgap between adj. comb fingers (finger_gap), um	3	
Rotor spoke width (rotor_spoke_width), um (rspokew)	12	
Stator spoke width (stator_spoke_width), um (sspokew)	14	
Gap between stator spokes at radius (stator_spoke_gap), um (sspokeg)	5	
Finger overlap as a fraction of length (finger_overlap), (trsovlp)	0.3	
Name of cell	rcdm1	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Rotary Side Drive Elements (S\_RSDM\_1, S\_RSDM\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a rotary side drive on the first or second structural layer. If the offset is set to zero, the first rotor tooth will be aligned with the first stator electrodes.

The rotary side drive designed on the first structural layer (S\_RSDM\_1) has a corresponding symbol (S\_RSDM\_1\_M\_PHI) . The rotary side drive designed on the second structural layer (S\_RSDM\_2) has a corresponding schematic (S\_RSDM\_2\_M\_PHI).

### Parameter List

The following table provides the rotary side drive parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Inner radius of rotor ring	rring	50 $\mu\text{m}$	rotor_ring_inner_radius
Inner radius of rotor tooth	rri	60 $\mu\text{m}$	rotor_pole_inner_radius

Description	Layout Parameter Name	Default Value	Schematic parameter name
Outer radius of rotor tooth	rro	150 $\mu\text{m}$	rotor_pole_outer_radius
Inner radius of stator electrode	rsi	155 $\mu\text{m}$	stator_pole_inner_radius
Outer radius of stator electrode	rso	200 $\mu\text{m}$	stator_pole_outer_radius
Angular width of rotor pole	phirp	18 degrees	rotor_pole_angular_width
Angular gap between adjacent rotor teeth	phirg	27 degrees	rotor_poles_angular_gap
Angular width of stator pole	phisp	18 degrees	stator_pole_angular_width
Angular gap between adjacent stator poles	phisg	12 degrees	stator_poles_angular_gap
Angular offset of rotor	roffset	0 degrees	rotor_stator_angular_offset

**S-Edit Symbol Name**     S\_RSDM\_1\_M\_PHI (for poly1 layer), S\_RSDM\_2\_M\_PHI (for poly2 layer)

## Rotary Side Drive Elements

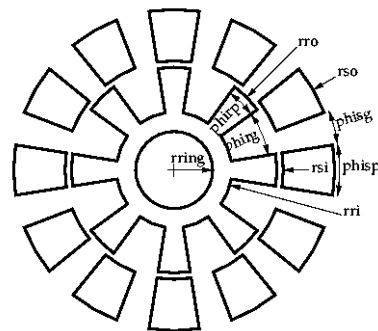
**Rotary Side Drive Parameters [S\_RSDM\_1]**

Inner radius of rotor ring (rotor_ring_inner_radius), um (rri)	50	OK Cancel
Inner radius of rotor tooth (rotor_pole_inner_radius), um (ri)	60	
Outer radius of rotor tooth (rotor_pole_outer_radius), um (ro)	150	
Inner radius of stator electrode (stator_pole_inner_radius), um	155	
Outer radius of stator electrode (stator_pole_outer_radius), um (rs)	200	
Angular width of rotor pole (rotor_pole_angular_width), degrees (phirp)	18	
Angular gap between adj. rotor teeth (rotor_poles_angular_gap), degrees	27	
Angular width of stator pole (stator_pole_angular_width), degrees (phisg)	18	
Angular gap between adj. stator poles (stator_poles_angular_gap), degrees	12	
Angular offset of rotor (rotor_stator_angular_offset), degrees	0	
Name of cell	rsdm1	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Harmonic Side Drive Elements (S\_HSDM\_1, S\_HSDM\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a harmonic side drive on the first or second structural layer. A central bearing (bearing1, bearing2) has to be added to complete the harmonic or wobble motor.

The rotary side drive designed on the first structural layer (S\_HSDM\_1) has a corresponding symbol (S\_HSDM\_1\_M\_PHI) . The rotary side drive designed on the second structural layer (S\_HSDM\_2) has a corresponding schematic (S\_HSDM\_2\_M\_PHI).

### Parameter List

The following table provides the harmonic side drive parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Inner rotor radius	rri	5 $\mu\text{m}$	rotor_inner_radius
Outer rotor radius	rro	60 $\mu\text{m}$	rotor_outer_radius

Description	Layout Parameter Name	Default Value	Schematic parameter name
Rotor ring width	rwidth	10 $\mu\text{m}$	rotor_ring_width
Stator inner radius	rsi	65 $\mu\text{m}$	stator_inner_radius
Stator outer radius	rso	125 $\mu\text{m}$	stator_outer_radius
Stator pole angle	phisp	18 degrees	stator_pole_angle
Angular gap between poles	phisg	12 degrees	stator_poles_angular_gap

**S-Edit Symbol Name**      S\_HSDM\_1\_M\_PHI (for poly1 layer), S\_HSDM\_2\_M\_PHI (for poly2 layer)

## Harmonic Side Drive Elements

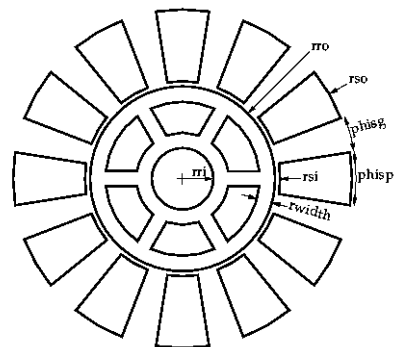
**Harmonic Side Drive Parameters [S\_HSDM\_1]**

Inner rotor radius (rotor_inner_radius), um (ri)	<input type="text" value="5"/>	<input type="button" value="OK"/>
Outer rotor radius (rotor_outer_radius), um (ro)	<input type="text" value="60"/>	<input type="button" value="Cancel"/>
Rotor ring width (rotor_ring_width), um (rwidth)	<input type="text" value="10"/>	
Stator inner radius (stator_inner_radius), um (rsi)	<input type="text" value="65"/>	
Stator outer radius (stator_outer_radius), um (rso)	<input type="text" value="125"/>	
Stator pole angle (stator_pole_angle), degrees (phisip)	<input type="text" value="18"/>	
Angular gap between poles (stator_poles_angular_gap), degrees (phisig)	<input type="text" value="12"/>	
Name of cell	<input type="text" value="hsdm1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration



# Passive Elements

## Journal Bearing Elements 1 (S\_JBEARG\_1)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a journal bearing intended to connect with a rotary element on the first structural layer. The shaft is anchored to the substrate, and the retaining cap on top of the shaft central to the bearing is formed on the second structural layer. The outside of the shaft on structural layer2 is one bearing surface while the inside of the rotor on structural layer1 is the second-bearing surface. The clearance between the two bearing surfaces is determined by the thickness of the second sacrificial layer used in the surface micromachining fabrication process. The radius of the shaft is set by the inner radius of the journal rotor and the second sacrificial layer thickness used in the process.

## Parameter List

The following table provides the journal bearing 1 parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Radius of cap of central shaft	rcap	8.5 $\mu\text{m}$	cap_radius
Inner radius of journal rotor	rinner	4.5 $\mu\text{m}$	rotor_inner_radius
Outer radius of journal rotor	router	15 $\mu\text{m}$	rotor_outer_radius

**S-Edit Symbol Name**     S\_JBEARG\_1

**S-Edit Test Schematic**     N/A

## Journal Bearing Elements 1

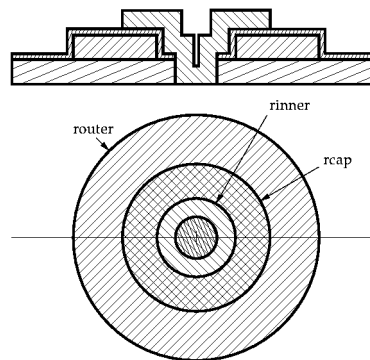
**Journal Bearing Parameters [S\_JBEARG\_1]**

Radius of cap of central shaft (cap_radius), um (rcap)	<input type="text" value="8.5"/>	<input type="button" value="OK"/>
Inner radius of journal rotor (rotor_inner_radius), um (rinner)	<input type="text" value="4.5"/>	<input type="button" value="Cancel"/>
Outer radius of journal rotor (rotor_outer_radius), um (router)	<input type="text" value="15"/>	
Name of cell	<input type="text" value="jbearg1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Journal Bearing Elements 2 (S\_JBEARG\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a journal bearing intended to connect with a rotary element on the second structural layer. The outside of the shaft on structural layer2 is one bearing surface while the inside of the rotor on structural layer1 is the second bearing surface. The clearance between the two bearing surfaces is determined by the thickness of the second sacrificial layer. The radius of the shaft is set by the inner radius of the journal rotor and the second sacrificial layer thickness used in the process. The rotor has an outer ring on structural layer2 that is mechanically connected to the rotary part of the bearing on structural layer1.

### Parameter List

The following table provides the journal bearing 2 parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Radius of cap of central shaft	r <sub>cap</sub>	8.5 $\mu\text{m}$	cap_rafius
Inner radius of journal rotor	r <sub>inner</sub>	4.5 $\mu\text{m}$	rotor_inner_radius

Description	Layout Parameter Name	Default Value	Schematic parameter name
Outer radius of journal rotor	router	15 $\mu\text{m}$	rotor_outer_radius

**S-Edit Symbol Name**     S\_JBEARG\_2

**S-Edit Test Schematic**     N/A

## Journal Bearing Elements 2

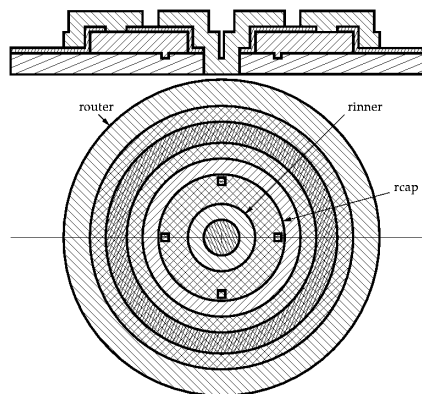
**Journal Bearing Parameters (S\_JBEARG\_2)**

Radius of cap of central shaft (cap_radius), um (rcap)	<input type="text" value="8.5"/>	<input type="button" value="OK"/>
Inner radius of journal rotor (rotor_inner_radius), um (rinner)	<input type="text" value="4.5"/>	<input type="button" value="Cancel"/>
Outer radius of journal rotor (rotor_outer_radius), um (router)	<input type="text" value="15"/>	
Name of cell	<input type="text" value="jbearg2"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Linear Crab Leg Suspension Elements - Type 1 (S\_LCLS\_1, S\_LCLS\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a linear crab leg suspension on the first or second structural layer. The local origin of the element is at the center of the shuttle mass. Actuators can be connected to the yokes on the shuttle mass.

### Parameter List

The following table provides linear crab leg suspension parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Length of beam1	lbeam1	30 $\mu\text{m}$	beam1_length
Width of beam1	wbeam1	20 $\mu\text{m}$	beam1_width
Length of beam2	lbeam2	75 $\mu\text{m}$	beam2_length
Width of beam2	wbeam2	8 $\mu\text{m}$	beam2_width

Description	Layout Parameter Name	Default Value	Schematic parameter name
Seperation between type 1 beams	beam1sep	70 $\mu\text{m}$	beams_separation
Width of shuttle	swidth	30 $\mu\text{m}$	shuttle_width
Length of shuttle	slength	100 $\mu\text{m}$	shuttle_lenth
Width of anchor support	wanchor	25 $\mu\text{m}$	anchor_width
Width of shuttle yoke	wsyoke	12 $\mu\text{m}$	shuttle_yoke_width
Length of shuttle yoke	lsyoke	98 $\mu\text{m}$	shuttle_yoke_length

**S-Edit Symbol Name**     S\_LCLS\_1\_M\_X, S\_LCLS\_2\_M\_X



## Linear Crab Leg Suspension Elements - Type 1

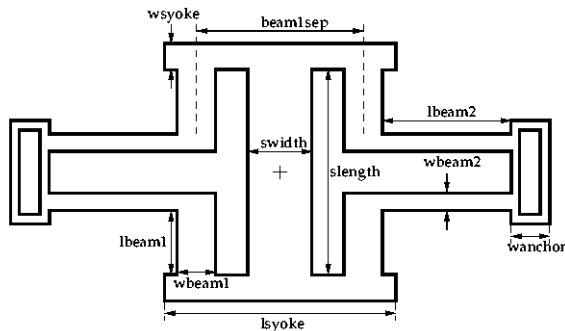
**Linear Crab Leg Suspension Parameters (\$\_LCLS\_1)**

Length of beam1 (beam1_length), um (lbeam1)	<input type="text" value="30"/>	<input type="button" value="OK"/>
Width of beam1 (beam1_width), um (wbeam1)	<input type="text" value="20"/>	<input type="button" value="Cancel"/>
Length of beam2 (beam2_length), um (lbeam2)	<input type="text" value="75"/>	
Width of beam2 (beam2_width), um (wbeam2)	<input type="text" value="8"/>	
Separation between type 1 beams (beams_separation), um (beam1sep)	<input type="text" value="70"/>	
Width of shuttle width (shuttle_width), um (swidth)	<input type="text" value="30"/>	
Length of shuttle (shuttle_length), um (slength)	<input type="text" value="100"/>	
Width of anchor support (anchor_width), um (wanchor)	<input type="text" value="25"/>	
Width of shuttle yoke (shuttle_yoke_width), um (wsyoke)	<input type="text" value="12"/>	
Length of shuttle yoke (shuttle_yoke_length), um (lsyoke)	<input type="text" value="98"/>	
Name of cell	<input type="text" value="lcls1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Linear Crab Leg Suspension Elements - Type 2 (S\_LCLSB\_1, S\_LCLSB\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a linear crab leg suspension on the first or second structural layer. The local origin of the element is at the center of the shuttle mass. Actuators can be connected to the yokes on the shuttle mass. Unlike lcls, this element is anchored at 4 points.

### Parameter List

The following table provides the linear crab leg suspension parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Length of beam1	lbeam1	30 $\mu\text{m}$	beam1_length
Width of beam1	wbeam1	20 $\mu\text{m}$	beam1_width
Length of beam2	lbeam2	75 $\mu\text{m}$	beam2_length

Description	Layout Parameter Name	Default Value	Schematic parameter name
Width of beam2	wbeam2	8 $\mu\text{m}$	beam2_width
Seperation between type 1 beams	beam1sep	70 $\mu\text{m}$	beams_separation
Width of shuttle	swidth	30 $\mu\text{m}$	shuttle_width
Length of shuttle	slength	100 $\mu\text{m}$	shuttle_length
Width of anchor support	wanchor	25 $\mu\text{m}$	anchor_width
Width of shuttle yoke	wsyoke	12 $\mu\text{m}$	shuttle_yoke_width
Length of shuttle yoke	lsyoke	98 $\mu\text{m}$	shuttle_yoke_length

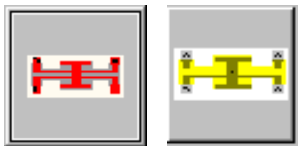
**S-Edit Symbol Name**     S\_LCLSB\_1\_M\_X, S\_LCLSB\_2\_M\_X

## Linear Crab Leg Suspension Elements - Type 2

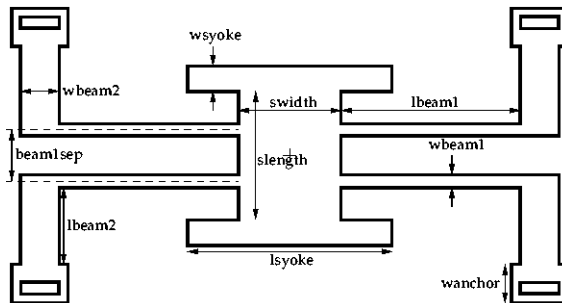
**Linear Crab Leg Suspension Parameters [S\_LCLSB\_1]**

Length of beam1 (beam1_length), um (lbeam1)	<input type="text" value="30"/>	<input type="button" value="OK"/>
Width of beam1 (beam1_width), um (wbeam1)	<input type="text" value="20"/>	<input type="button" value="Cancel"/>
Length of beam2 (beam2_length), um (lbeam2)	<input type="text" value="75"/>	
Width of beam2 (beam2_width), um (wbeam2)	<input type="text" value="8"/>	
Separation between type 1 beams (beams_separation), um (beam1sep)	<input type="text" value="70"/>	
Width of shuttle width (shuttle_width), um (swidth)	<input type="text" value="30"/>	
Length of shuttle (shuttle_length), um (slength)	<input type="text" value="100"/>	
Width of anchor support (anchor_width), um (wanchor)	<input type="text" value="25"/>	
Width of shuttle yoke (shuttle_yoke_width), um (wsyoke)	<input type="text" value="12"/>	
Length of shuttle yoke (shuttle_yoke_length), um (lsyoke)	<input type="text" value="98"/>	
Name of cell	<input type="text" value="lclsb1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Linear Folded Beam Suspension Elements (S\_LFBS\_1, S\_LFBS\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a linear folded beam suspension on the first or second structural layer. Actuators or other mechanical elements can be connected to the yokes at the ends of the shuttle mass.

### Parameter List

The following table provides the linear folded beam suspension parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Length of beam	lbeam	150 $\mu\text{m}$	flexure_length
Width of beam	wbeam	4 $\mu\text{m}$	flexure_width
Seperation between beams	beamsep	50 $\mu\text{m}$	beams_separation
Width of connecting bar	wbar	12 $\mu\text{m}$	truss_width
Width of shuttle	swidth	30 $\mu\text{m}$	shuttle_width

Description	Layout Parameter Name	Default Value	Schematic parameter name
Width of anchor support	wanchor	25 $\mu\text{m}$	anchor_width
Width of shuttle yoke	wsyoke	12 $\mu\text{m}$	shuttle_yoke_width
Length of shuttle yoke	lsyoke	98 $\mu\text{m}$	shuttle_yoke_length

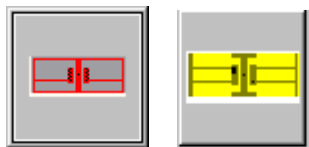
**S-Edit Symbol Name**     S\_LFBS\_1\_M\_X, S\_LFBS\_2\_M\_X

## Linear Folded Beam Suspension Elements

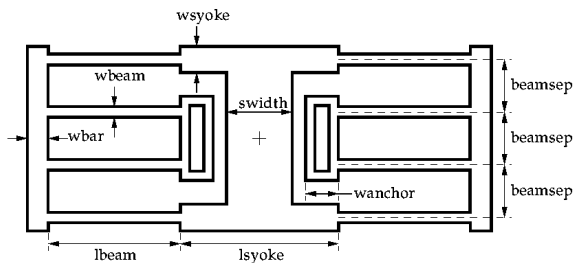
**Linear Folded Beam Suspension w/ Ground Plane Parameters [S\_LFBS\_1]**

Length of beam (flexure_length), um (lbeam)	150	OK Cancel
Width of beam (flexure_width), um (wbeam)	4	
Separation between beams (beams_separation), um (beamsep)	50	
Width of connecting bar (truss_width), um (wbar)	12	
Width of shuttle width (shuttle_width), um (swidth)	30	
Width of anchor support (anchor_width), um (wanchor)	25	
Width of shuttle yoke (shuttle_yoke_width), um (wsyoke)	12	
Length of shuttle yoke (shuttle_yoke_length), um (lsyoke)	98	
Name of cell	lfs1	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Dual Archimedean Spiral Spring Elements (S\_SPIRAL\_1, S\_SPIRAL\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates dual archimedean spiral springs on the first or second structural layer. A possible application of the spiral spring is in a torsional suspension system. Actuators or other mechanical elements can be connected to the rotor supports at the ends of the spiral spring. The length parameter of the spiral beam corresponds to the length of the central axis of the beam. The element parameters can be selected to obtain the electrical connect layer on the dielectric properties of the isolation layer.

### Parameter List

The following table provides the dual archimedean spiral spring parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Radius of support shaft	rshaft	10 $\mu\text{m}$	support_shaft_radius



Description	Layout Parameter Name	Default Value	Schematic parameter name
Starting radius of spiral beam	rinner	15 $\mu\text{m}$	spiral_initial_radius
Final radius of spiral beam	router	65 $\mu\text{m}$	spiral_final_radius
Length of each spiral beam	length	300 $\mu\text{m}$	spiral_length
Width of the spiral beam	width	2 $\mu\text{m}$	spiral_width
Outer radius of rotor support	rrotor	80 $\mu\text{m}$	rotor_outer_radius

**S-Edit Symbol Name**      S\_SPIRAL\_1\_M\_PHI, S\_SPIRAL\_2\_M\_PHI

## Dual Archimedean Spiral Spring Elements

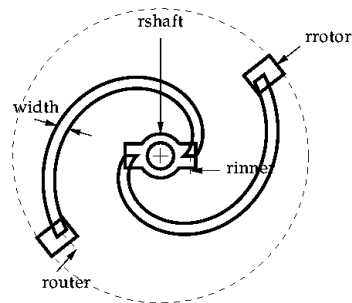
**Spiral Parameters (S\_SPIRAL\_1)**

Radius of support shaft (support_shaft_radius), um (rshaft)	<input type="text" value="10"/>	<input type="button" value="OK"/>
Starting radius of spiral beam (spiral_initial_radius), um	<input type="text" value="15"/>	<input type="button" value="Cancel"/>
Final radius of spiral beam (spiral_final_radius), um (router)	<input type="text" value="65"/>	
Length of each spiral beam (spiral_length), um	<input type="text" value="300"/>	
Width of the spiral beam (spiral_width), um (width)	<input type="text" value="2"/>	
Outer radius of rotor support (rotor_outer_radius), um (rrotor)	<input type="text" value="80"/>	
Name of cell	<input type="text" value="spiral1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

# Test Elements

## Area-Perimeter Dielectric Isolation Test Structure Element (S\_APTEST\_1)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates an area-perimeter test structure that can be used to test the dielectric properties of the isolation layer between the first electrical connect layer and the substrate. It can also be used to measure the resistance of the first electrical contact layer. Probe pads are included in the structure to allow electrical probing for measurements. An electrical connection to the conductive substrate is required for dielectric measurements.

### Parameter List

The following table provides the area-perimeter dielectric isolation test structure parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Width of electrical connect wire	width	50 $\mu\text{m}$	anchor_width
Serpentine height	height	940 $\mu\text{m}$	serpentine_height
Serpentine wavelength	length	60 $\mu\text{m}$	serpentine_half_wavelength
Number of wavelengths	nw	10 $\mu\text{m}$	number_of_wavelengths

**S-Edit Symbol Name**     N/A

## Area-Perimeter Dielectric Isolation Test Structure Element

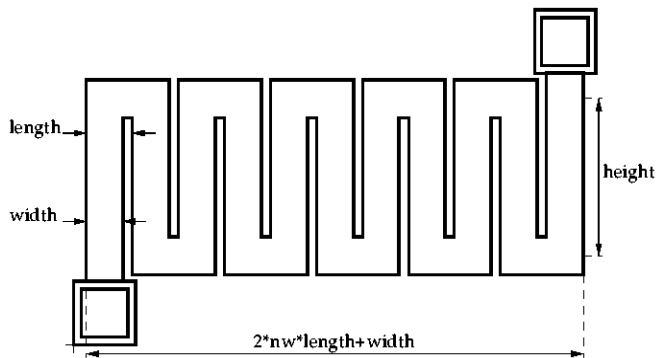
**Area Perimeter Parameters [S\_APTTEST\_1]**

Width of electrical connect wire (anchor_width), um (width)	<input type="text" value="50"/>	<input type="button" value="OK"/>
Serpentine height (serpentine_height), um (height)	<input type="text" value="340"/>	<input type="button" value="Cancel"/>
Serpentine half wave length (serpentine_half_wavelength), um (length)	<input type="text" value="60"/>	
Number of wavelengths (number_of_wavelengths), um (nw)	<input type="text" value="10"/>	
Name of cell	<input type="text" value="aptest"/>	

Layout Parameter entry dialog



Layout Palette Button



Layout Parameter Illustration

## Crossover Test Structure Element - Type 1 (S\_COTEST\_1)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a crossover test structure that can be used to test electrical interconnection using bridges on structural layers 1 and 2 to cross over lines on the first electrical interconnect layer. The wires are anchored to the substrate except at the bridges.

### Parameter List

The following table provides the crossover test structure (type 1) parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Wire width of electrical connect layer	p0width	10 $\mu$ m	elec_layer_line_width

Description	Layout Parameter Name	Default Value	Schematic parameter name
Wire width of first structural layer	p1width	12 $\mu\text{m}$	first_struct_layer_line_width
Wire width of second structural layer	p2width	12 $\mu\text{m}$	second_struct_layer_line_width

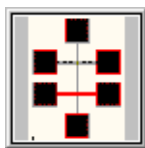
**S-Edit Symbol Name**      N/A

## Crossover Test Structure Element - Type 1

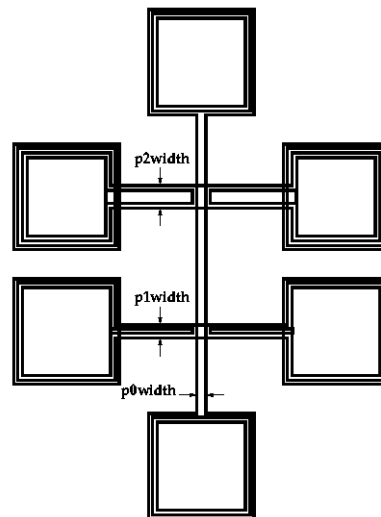
**Multi-layer Cross Over Parameters (S\_CDTEST\_1)**

Wire width of elect. connect layer (elect_layer_line_width), um (p0width)	<input type="text" value="10"/>	<input type="button" value="OK"/>
Wire width of first structural layer (first_layer_line_width), um (p1width)	<input type="text" value="12"/>	<input type="button" value="Cancel"/>
Wire width of second structural layer (second_layer_line_width), um	<input type="text" value="12"/>	
Name of cell	<input type="text" value="cotest1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration



## Crossover Test Structure Element - Type 2 (S\_COTEST\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a crossover test structure that can be used to test electrical interconnection using bridges on structural layer2 to cross over lines on the first electrical interconnect layer. The wires are anchored to the substrate except at the bridges.

### Parameter List

The following table provides the crossover test structure (type 2) parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Wire width of first structural layer	p1width	10 $\mu\text{m}$	first_struct_layer_line_width
Wire width of second structural layer	p2width	12 $\mu\text{m}$	second_struct_layer_line_width

<b>S-Edit Symbol Name</b>	N/A
---------------------------	-----

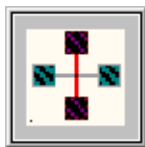


## Crossover Test Structure Element - Type 2

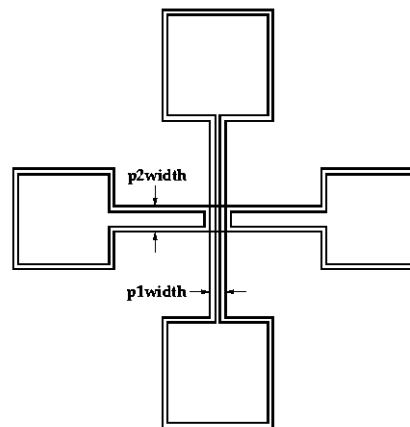
**Crossover Parameters [S\_COTEST\_2]**

Wire width of first structural layer (first_layer_line_width), um (p1 width)	<input type="text" value="10"/>	<input type="button" value="OK"/>
Wire width of second structural layer (second_layer_line_width), um	<input type="text" value="12"/>	<input type="button" value="Cancel"/>
Name of cell	<input type="text" value="cotest2"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Euler Column (Doubly Supported Beam) Elements (S\_EUBEAM\_1, S\_EUBEAM\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a doubly supported beam test structure on the first or second structural layer. This element can be used to estimate the residual strain in a film with a compressive residual strain. Generally, an array of beams with varying lengths is used to determine the critical buckling length for the residual strain in the structural layer of interest. Hence, the name, Euler columns, for these test structures. The beam parameters are chosen to set the critical buckling strain of the beam and hence the residual compressive strain that it would detect. If the thickness of the structural layer used is larger than the width of the beam, lateral buckling will occur; i.e., buckling in the plane of the wafer. Otherwise, buckling will occur out of the plane of the wafer.

### Parameter List

The following table provides the Euler column parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Length of doubly supported beam	blength	200 $\mu\text{m}$	beam_length
Width of doubly supported beam	bwidth	20 $\mu\text{m}$	beam_width
Size of anchor supports	asize	30 $\mu\text{m}$	anchor_size

**S-Edit Symbol Name**      N/A

## Euler Column (Doubly Supported Beam) Elements

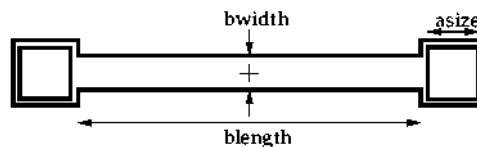
**Euler Column Parameters [S\_EUBEAM\_1]**

Length of doubly supported beam (beam_length), um (blength)	<input type="text" value="200"/>	<input type="button" value="OK"/>
Width of doubly supported beam (beam_width), um (bwidth)	<input type="text" value="20"/>	<input type="button" value="Cancel"/>
Size of anchor supports (anchor_size), um (asize)	<input type="text" value="30"/>	
Name of cell	<input type="text" value="dsbeam1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Array of Euler Column Elements (S\_EUBEAMS\_1, S\_EUBEAMS\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a set of doubly supported beam test structures on the first or second structural layer. This element can be used to estimate the residual strain in a film with a compressive residual strain. The element uses the residual strain range and step size specified to determine the beam lengths of the array of doubly supported beams. The lengths are chosen such that the critical strain that the beams can support before buckling corresponds to the desired value of residual strain to be detected. Euler buckling criterion for the compressive strain in the film is used.

### Parameter List

The following table provides the array of Euler column parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Minimum residual strain	e0min	0.0005 $\mu\text{m}$	residual_strain_minimum
Maximum residual strain	e0max	0.0025 $\mu\text{m}$	residual_strain_maximum

Description	Layout Parameter Name	Default Value	Schematic parameter name
Residual strain by step	dele0	0.00025 $\mu\text{m}$	residual_strain_step
Width of doubly supported beam	bwidth	20 $\mu\text{m}$	beams_width
Size of anchor supports	asize	30 $\mu\text{m}$	anchor_size
Thickness of beam structural layer	heigh	2 $\mu\text{m}$	beam_thickness

**S-Edit Symbol Name**     N/A



## Array of Euler Column Elements

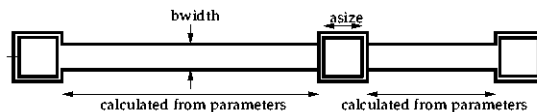
**Array of Euler Column Parameters (S\_EUBEAMS\_1)**

Minimum residual strain (residual_strain_minimum), um	<input type="text" value="0.0005"/>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>
Maximum residual strain (residual_strain_maximum), um	<input type="text" value="0.0025"/>	
Residual strain by step (residual_strain_step), um	<input type="text" value="0.00025"/>	
Width of doubly supported beam (beam_width), um (bwidth)	<input type="text" value="20"/>	
Size of anchor supports (anchor_size), um (asize)	<input type="text" value="30"/>	
Thickness of beam structural layer (beam_thickness), um	<input type="text" value="2"/>	
Name of cell	<input type="text" value="dsbeams1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

The array of Euler columns depend on the following equation:

$$G_c = \frac{x^2 h^2}{3L^2}$$

where L is the length, and h is the height of the beam (the minimum of width and height parameters specified). The anchored ends of the beam are considered ideal clamped ends and the elasticity of the supports is not modeled. The beam parameters are chosen to set the critical buckling strain of the beam and hence the residual compressive strain that it would detect. If the thickness of the structural layer used is larger than the width of the beam, lateral buckling will occur.

## Guckel Ring Test Structure Elements (S\_GURING\_1, S\_GURING\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a single “Guckel” ring test structure on the first or second structural layer. These ring structures can be used to estimate the residual strain in a film with tensile residual strain. An array of rings with different radii are used to estimate the critical radius at which buckling occurs in the cross beam of the test structure and hence infer the tensile residual stress present in the structural film. The ring parameters are chosen to set the critical buckling strain of the cross beam and hence the

residual tensile strain that it would detect. If the thickness of the structural layer used is larger than the width of the cross beam, lateral buckling will occur; i.e., buckling in the plane of the wafer. Otherwise, buckling will occur out of the plane of the wafer.

## Parameter List

The following table provides the Guckel ring test structure parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Mean radius of ring	radius	200 $\mu\text{m}$	ring_mean_radius
Width of ring	bring	20 $\mu\text{m}$	ring_width
Width of cross beam	bbeam	10 $\mu\text{m}$	cross_beam_width
Size of anchor supports	asize	30 $\mu\text{m}$	anchor_size

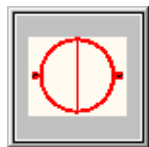
**S-Edit Symbol Name**      N/A

## Guckel Ring Test Structure Elements

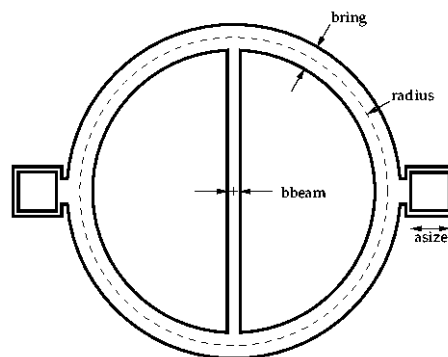
**Guckel Ring Parameters (S\_GURING\_1)**

Mean radius of ring (ring_mean_radius), um (radius)	<input type="text" value="200"/>	<input type="button" value="OK"/>
Width of ring (ring_width), um (bring)	<input type="text" value="20"/>	<input type="button" value="Cancel"/>
Width of cross beam (cross_beam_width), um (bbeam)	<input type="text" value="10"/>	
Size of anchor supports (anchor_size), um (asize)	<input type="text" value="30"/>	
Name of cell	<input type="text" value="guring1"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Array of Guckel Ring Elements (S\_GURINGS\_1)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates an array of “Guckel” ring test structures on the first or second structural layer. These ring structures can be used to estimate the residual strain in a film with tensile residual strain. The ring parameters are calculated for the critical strain values desired using a mechanical model of the test structure. If the thickness of the structural layer used is larger than the width of the cross beam, lateral buckling will occur; i.e., buckling in the plane of the wafer. Otherwise, buckling will occur out of the plane of the wafer.

### Parameter List

The following table provides the array of Guckel ring parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Minimum residual strain	e0min	0.0005	residual_strain_minimum
Maximum residual strain	e0max	0.0025	residual_strain_maximum

Description	Layout Parameter Name	Default Value	Schematic parameter name
Residual strain step size	dele0	0.00025	residual_strain_step
Width of ring	bring	20 $\mu\text{m}$	ring_width
Width of cross beam	bbeam	10 $\mu\text{m}$	cross_beam_width
Size of anchor supports	asize	40 $\mu\text{m}$	anchor_size
Thickness of structural layer	heighth	2 $\mu\text{m}$	ring_thickness
Poisson's ratio for structural film	nu	0.23	poissons_ratio

**S-Edit Symbol Name**      N/A

## Array of Guckel Ring Elements

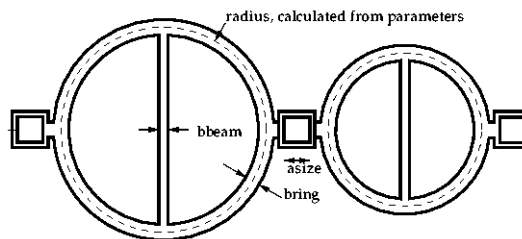
**Array of Guckel Rings Parameters (S\_GURINGS\_1)**

Minimum residual strain (residual_strain_minimum)	0.0005	OK
Maximum residual strain (residual_strain_maximum)	0.0025	Cancel
Residual strain step size (residual_strain_step)	0.00025	
Width of ring (ring_width), um (bring)	20	
Width of cross beam (cross_beam_width), um (bbeam)	10	
Size of anchor supports (anchor_size), um (asize)	40	
Thickness of structural layer (ring_thickness), um	2	
Poisson's ratio for structural film (poissons_ratio)	0.23	
Name of cell	gurings1	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Multilayer Pad Element (S\_PAD\_1)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a pad for wafer probe or wire bond purposes. It has a stack of layers electrically connecting the first electrical connect layer, first structural layer, first structural layer, second structural layer, and the second (and final) electrical layer.

### Parameter List

The following table provides the multilayer pad parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Pad width	padw	100 $\mu$ m	pad_width

**S-Edit Symbol Name**     N/A



## Multilayer Pad Element

**Multi-Layer Parameters (S\_PAD\_1)**

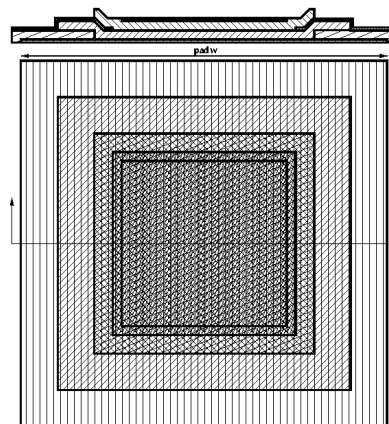
Pad width (pad\_width), um (padw)

Name of cell

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

# Resonator Elements

## Plate (S\_PLATE\_1)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Macro **MEMSLIB.DLL**

### Description

Generates a plate on the Poly1 layer.

### Parameter List

The following table provides the plate parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Plate width	width	200 $\mu\text{m}$	plate_width
Plate length	length	200 $\mu\text{m}$	plate_length

**S-Edit Symbol Name**    S\_PLATE\_1\_M\_X



## Plate

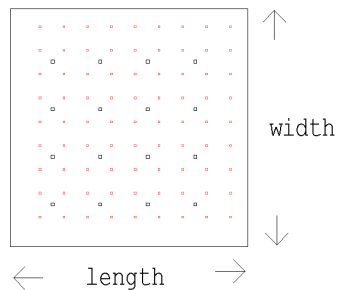
**Rectangular Parameters (S\_PLATE\_1)**

Plate width (plate_width), um (width)	<input type="text" value="200"/>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>
Plate length (plate_length), um (length)	<input type="text" value="200"/>	
Name of cell	<input type="text" value="plate1"/>	
Name of instance	<input type="text" value="PlateInst"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Comb Drive (S\_LCOMB\_3)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Cell **comb** / Macro **MEMSLIB.DLL**

### Description

Generates a comb drive on the Poly1 layer.

### Parameter List

The following table provides the comb drive parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Finger width	width	4 $\mu\text{m}$	width
Finger length	length	40 $\mu\text{m}$	length
Finger gap	gap	3 $\mu\text{m}$	gap
Finger overlap	overlap	15 $\mu\text{m}$	overlap
Number of gaps	ng	20	number_of_gaps

**S-Edit Symbol Name**     S\_LCOMB\_3\_M\_X



## Comb Drive (comb)

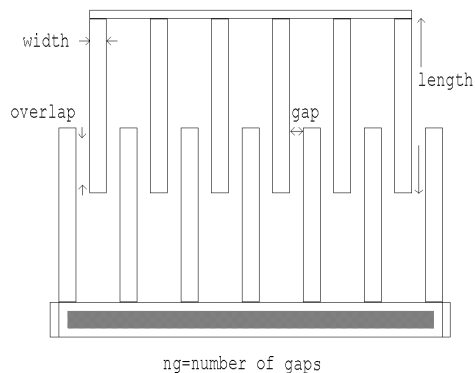
**Linear Comb Parameters (S\_LCOMB\_3)**

Finger width (finger_width), um (width)	<input type="text" value="4"/>	<input type="button" value="OK"/>
Finger length (finger_length), um (length)	<input type="text" value="40"/>	<input type="button" value="Cancel"/>
Finger gap (finger_gap), um (gap)	<input type="text" value="3"/>	
Finger overlap (finger_overlap), um (overlap)	<input type="text" value="15"/>	
Number of gaps (number_of_gaps), ng	<input type="text" value="20"/>	
Name cell	<input type="text" value="lcomb3"/>	
Name of instance	<input type="text" value="CombInst"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Folded Spring (S\_LFBS\_3)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Cell **fspring** / Macro **MEMSLIB.DLL**

### Description

Generates a folded spring on the Poly1 layer.

### Parameter List

The following table provides the folded springs parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Beam width	width	2 $\mu\text{m}$	flexure_width
Beam length	length	200 $\mu\text{m}$	flexure_length
Outer gap	inner gap	10 $\mu\text{m}$	flexure_outer_gap
Inner gap	outer gap	10 $\mu\text{m}$	flexure_inner_gap

**S-Edit Symbol Name**    S\_LFBS\_3\_M\_X





## Folded Spring

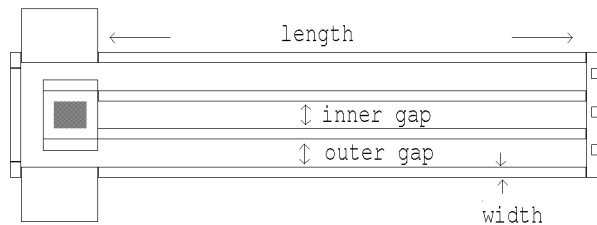
**Linear Folded Parameters [S\_LFBS\_3]**

Beam width (flexure_width), um (width)	<input type="text" value="2"/>	<input type="button" value="OK"/>
Beam length (flexure_length), um (length)	<input type="text" value="200"/>	<input type="button" value="Cancel"/>
Outer gap (flexure_outer_gap), um (outer gap)	<input type="text" value="10"/>	
Inner gap (flexure_inner_gap), um (inner gap)	<input type="text" value="10"/>	
Name of cell	<input type="text" value="lfbs3"/>	
Name of Instance	<input type="text" value="SpringInst"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Ground Plate (S\_GDPLATE\_1)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Cell **groundplate** / Macro **MEMSLIB.DLL**

**Description** Generates a ground plate on the Poly0 layer.

### Parameter List

The following table provides the ground plate parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Ground plate width	width	300 $\mu\text{m}$	plate_width
Ground plate length	length	600 $\mu\text{m}$	plate_length

**S-Edit Symbol Name** S\_GDPLATE\_1

**S-Edit Test Schematic** N/A

## Ground Plate

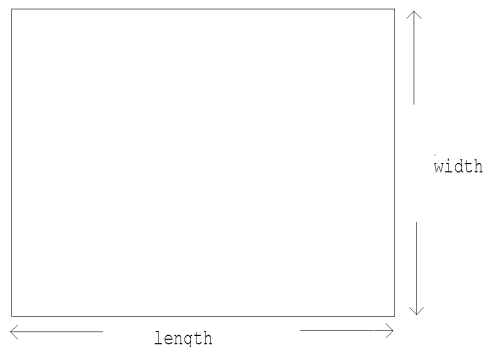
**Rectangular Plane Parameters [\$\_GDPLATE\_1\$]**

Grnd plate width (plate_width), um (width)	<input type="text" value="300"/>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>
Grnd plate length (plate_length), um (length)	<input type="text" value="600"/>	
Name of cell	<input type="text" value="gdplate1"/>	
Name of instance	<input type="text" value="GroundPlateInst"/>	

### Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

## Bonding Pad (S\_PAD\_2)

*S-Edit:* File **MEMSLIB.SDB**

*L-Edit:* File **MEMSLIB.TDB** / Cell **bpad** / Macro **MEMSLIB.DLL**

**Description** Generates a bonding pad.

### Parameter List

The following table provides the bonding pad parameters, their values and descriptions.

Description	Layout Parameter Name	Default Value	Schematic parameter name
Pad width	width	100 $\mu\text{m}$	pad_width
Pad length	length	100 $\mu\text{m}$	pad_length

**S-Edit Symbol Name** S\_PAD\_2

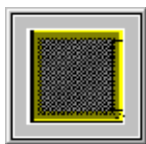
**S-Edit Test Schematic** N/A

## Bonding Pad

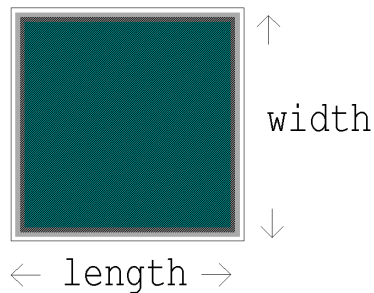
**Bonding Parameters [S\_PAD\_2]**

Pad width (pad_width), um (width)	<input type="text" value="100"/>	<input type="button" value="OK"/> <input type="button" value="Cancel"/>
Pad length (pad_length), um (length)	<input type="text" value="100"/>	
Name of cell	<input type="text" value="pad2"/>	
Name of instance	<input type="text" value="PadInst"/>	

Layout Parameter Entry Dialog Box



Layout Palette Button



Layout Parameter Illustration

# 16 Technology Setup

---

▪ Introduction	615
▪ MCNC MUMPs	616
▪ Analog Devices/MCNC iMEMS	619
▪ Sandia ITT	620
▪ MOSIS/CMU	621
▪ MOSIS/NIST	622

# Introduction

For your convenience, we have included technology setup information for several processes: MCNC(MUMPS), Sandia(ITT), ADI/MCNC (iMEMS), MOSIS/CMU and MOSIS/NIST (SCNAMEMS). The process setup information includes design rules, layer definitions, extraction rules, process definitions, model parameter values, and macros.



# MCNC MUMPs

**Note**

---

See <http://mems.mcnc.org/mumps.html> for the most up-to-date information.

---

The Multi-User MEMS Processes (MUMPs) is a program designed to provide low-cost access to MEMS technology. Hundreds of users from the domestic industry, government and academic communities have used MUMPs to access MEMS. And beginning with MUMPs run #26, access was made available worldwide to the international MEMS community.

MUMPs provides low-risk opportunities for small business and industry to access the prototype-to-commercialization pathway. Past participants have created a wide range of devices using MUMPs, including accelerometers, micro optical components, actuators, motors and many others.

MUMPs is a three-layer polysilicon surface micro-machining process designed to be as general as possible to provide maximum user flexibility. Polysilicon is used as the structural material, deposited oxide (PSG) as the sacrificial material, and silicon nitride for electrical isolation from the substrate. The process is derived from work performed by the Berkeley Sensors and Actuators Center at the University of California, Berkeley.

The process is different from most customized surface micromachining processes in that it is designed to be as general as possible, and to be capable of supporting many designs on a single silicon wafer. Since the process was not optimized with the purpose of fabricating any one specific device, the thicknesses of the structural and sacrificial layers were chosen to suit most users, and the layout design rules were chosen conservatively to guarantee the highest possible yield.

## Process Cross-Section

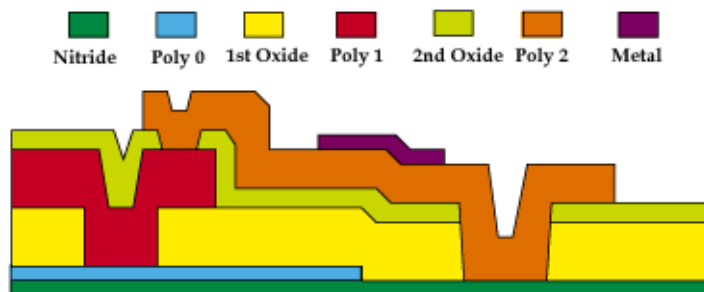
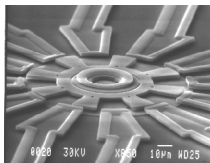
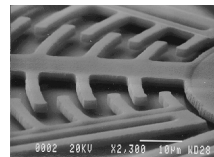


Figure 129: Cross-Section of the MUMPs process

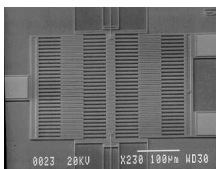
## Device Examples



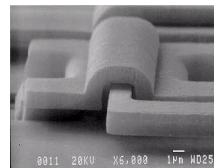
**Rotary Side-drive Motor**



**Rotary Comb-drive**



**Linear Comb-drive Resonator**



**Hinge**

## Analog Devices/MCNC iMEMS

**Note**

---

See <http://imems.mcnc.org/> for the most up-to-date information.

---

MEMS stands for Integrated Micro machined-mechanical Systems. This process supports BiCMOS with MEMS surface micromachining with polysilicon as the structural layer. Please refer to the website for more details.

# Sandia ITT

**Note**

---

See <http://www.mdl.sandia.gov/micromachine/integrated.html> for the most up-to-date information.

---

EXPLANATIONS TO BE ADDED

# MOSIS/CMU

**Note**

---

See <http://www.ece.cmu.edu/~mems/cmos-mems.html> for the most up-to-date information.

---

EXPLANATIONS TO BE ADDED

# MOSIS/NIST

**Note**

---

See <http://www.mosis.org/New/Technical/Designsupport/nist-mems-1.html> for the most up-to-date information.

---

This process enables the fabrication of MEMS structures using standard CMOS technology and a maskless post processing step. Please refer to the website for more details.

# 17 Process Definition

---

▪ Introduction	624
▪ Process Steps	628
▪ ProcessInfo	628
▪ Wafer	630
▪ Deposit	634
▪ Etch	647
▪ MechanicalPolish	659
▪ ImplantDiffuse	664
▪ Grow	669
▪ Process Definition Example: MUMPs	676



## Introduction

The 3D Modeler ascertains the three dimensional (3D) characteristics of the designed device from fabrication process definition information in combination with L-Edit mask layout. The process information is contained in a *process definition* (.pdt) file.

The process definition file is a text file that describes, in sequence, the geometric effect of the fabrication steps used to build a device. Process definitions are parameterized in geometric terms, that is, in terms such as etch depths and etch angles, not in processing terms, such as time of immersion or ambient temperature.

Process information can be entered into a process definition file and imported into L-Edit, or manually entered using the MEMS Pro **Edit Process Definition** dialog. From the **Edit Process Definition** dialog, the information can be stored in a process definition file and/or stored as part of the layout in the Tanner Database (.tdb) file.

Many designs use standard foundry processes for which process definition files have already been written. MEMS Pro includes complete process definition files in design kits for major MEMS foundries, including MCNC (MUMPS), Sandia (M3S), ADI (IMEMS), and MOSIS (NIST). Details about design kits can be found in chapter Technology Setup on page 343.

Process definitions sequentially list process commands and their parameters. All process parameters must appear between curly braces { }. The style for each entry appears in the sample code below.

```
Command={  
    Parameter=setting  
    Parameter=setting  
    Parameter="text"  
    Parameter=real number  
    Parameter=setting  
    Parameter="text"  
    Label="label text"  
    Comment="comment text"  
}
```

Although the parameters order may vary, the parameters list must be complete. An actual process definition file appears in its entirety at the end of this chapter in the section entitled Process Definition Example: MUMPs on page 676.

As you read the process command descriptions that appear later in the chapter, you will note that parameters require specific inputs, such as text, numbers, or switches. Parameter types and the values that may be entered for them are listed in the table below. In general, character strings (for example, *S* and *layer*) must be enclosed in quotation marks. Parameter types are case-sensitive. The 3D Modeler will not recognize *top* as *TOP*. In general, numbers (for example, *I* and *P*) do not require quotation marks. Switches (for example, **TOP**, **BOT**, and **microns**), do not require quotation marks, and they must be entered exactly as they are given in the reference table. The numeric parameters of type *I*, *P* and *R*

are in the units specified in the ProcessInfo step. Do not enter the unit when setting these parameters' values. For details on specifying units, see ProcessInfo on page 628.

### Permissible Values for Process Parameter Types

<i>Type</i>	<i>Permissible Value(s)</i>
A	Angle in degrees. Any value between 0 and 90.
I	An integer. A whole number between -2147483648 and 2147483648.
P	A positive real number. Any positive decimal number.
R	A real number. Any decimal number.
S	A string. Text that must be enclosed in quotation marks.
percent	A decimal number between 0 and 100.
layer	Any valid L-Edit layer name. Text enclosed in quotation marks.
unit	Any of the following units of length: <b>microns</b>   <b>millimeters</b>   <b>centimeters</b>   <b>mils</b>   <b>inches</b>   <b>lambda</b>   <b>other</b> .
face	Any of the following three options: <b>TOP</b>   <b>BOT</b>   <b>TOPBOT</b> . Do not enclose in quotation marks.
dtype	Any of the following three options: <b>CONFORMAL</b>   <b>SNOWFALL</b>   <b>FILL</b> . Do not enclose in quotation marks.

<i>Type</i>	<i>Permissible Value(s)</i>
scf	The scf type characterizes the <b>Scf</b> parameter. It accepts any decimal fraction (a real number from 0.0 to 1.0) enclosed in quotation marks. The character value <b>c</b> , enclosed in quotation marks, is also an acceptable input for <b>scf</b> type.
etype	Any of the following three options: <b>SURFACE</b>   <b>BULK</b>   <b>SACRIFICIAL</b> . Do not enclose in quotation marks.
emask	Any of the following two options: <b>INSIDE</b>   <b>OUTSIDE</b> . Do not enclose in quotation marks.

In the following pages, the parameters for each step and the options available for the designer are described in details.

## Process Steps

Process definitions are concatenated from strings of **Process Steps**. These steps, or commands, appear in the file in the same order as they occur in the actual fabrication processing.

Below, the syntax of each **Process Step** is described, followed by a description of the step, its uses, and its parameters.

### ProcessInfo

The syntax for the **ProcessInfo** step is presented below, followed by an example containing valid entries for its parameters.

#### *Syntax*

```
ProcessInfo={  
    Name=S  
    Version=S  
    Unit=unit  
}
```

## Example

```
ProcessInfo = {
    Name = "MUMPS"
    Version = "1.0beta"
    Unit = microns
}
```

## Description

The **ProcessInfo** command identifies the emulated fabrication process by **Name**, **Version**, and **Unit**. It must be the first block of your process definition file. Parameters for **ProcessInfo** are described below.

<i>Parameter</i>	<i>Description</i>
<b>Name</b>	Any valid process name enclosed in quotation marks is an acceptable entry for <b>Name</b> .
<b>Version</b>	Any valid version number enclosed in quotation marks is an acceptable entry for <b>Version</b> .
<b>Unit</b>	Any of the following length units may be entered for <b>Unit</b> : <b>microns</b>   <b>millimeters</b>   <b>centimeters</b>   <b>mils</b>   <b>inches</b>   <b>lambda</b>   <b>other</b> . Do not enclose the entry in quotation marks.

## Wafer

The syntax for the **Wafer** command is presented below, followed by an example containing valid entries for its parameters.

### *Syntax*

```
Wafer={  
    WaferID=S  
    MaskName=layer  
    Thickness=P  
    Target=layer  
    Label=S  
    Comment=S  
}
```

### *Example*

```
Wafer={  
    MaskName="substrate"  
    Thickness=5  
    WaferID="w1"  
    Target="substrate"  
    Label="Wafer"  
    Comment="Wafer"  
}
```

## *Description*

A **Process Step** can be applied only once a wafer has been identified. The **Wafer** command establishes a wafer and assigns it a name.

Multiple wafers are not supported in MEMS Pro Version 3. If multiple **Wafer** steps exist, only the first **Wafer** command will be used. A warning will be issued as the extraneous **Wafer** commands are encountered by the 3D Modeler.

Parameters for the **Wafer** command are described below.

<i>Parameter</i>	<i>Description</i>
<b>WaferID</b>	This optional parameter identifies the new <b>Wafer</b> . Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.



<i>Parameter</i>	<i>Description</i>
<b>MaskName</b>	<p>The geometry drawn on the <b>MaskName</b> layer defines the extent of the wafer. The boundary of the mask is usually defined by a box, but any drawing object may be used, including circles and curved polygons. Multiple objects that are not touching can also be drawn on the <b>MaskName</b> layer to define the <b>Wafer</b> extent. Objects that extend past the drawn wafer extent will be truncated.</p> <p>If no closed curve is drawn on the <b>MaskName</b> layer, its extent will be set to 110% of the minimum bounding box of the layout on all other masks.</p> <p>Any valid L-Edit layer name enclosed in quotation marks is an acceptable input for <b>MaskName</b>.</p>
<b>Thickness</b>	<p>Vertical dimension of the <b>Wafer</b>. Any positive decimal number is an acceptable input for <b>Thickness</b>.</p>
<b>Target</b>	<p>This parameter specifies the 3D model rendering characteristics of the <b>Wafer</b>. You may enter any valid L-Edit layer name as text enclosed in quotation marks for <b>Target</b>. <b>Target</b> and <b>Maskname</b> may be set to the same layer. For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 114.</p>

<i>Parameter</i>	<i>Description</i>
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.
<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.

## Deposit

The syntax for the **Deposit** command is presented below, followed by an example containing valid entries for its parameters.

### *Syntax*

```
Deposit={  
    WaferID=S  
    DepositType=dtype  
    Face=face  
    LayerName=layer  
    Thickness=P  
    Scf=scf  
    Target=layer  
    Label=S  
    Comment=S  
}
```

### *Example*

```
Deposit={  
    DepositType=CONFORMAL  
    Face=TOP  
    LayerName="Poly0"  
    Thickness=.5  
    Scf="c"
```

```
    WaferID="w1"  
    Target="Poly0"  
    Label="Deposit Poly0"  
    Comment="Deposit Poly0"  
}
```

### *Description*

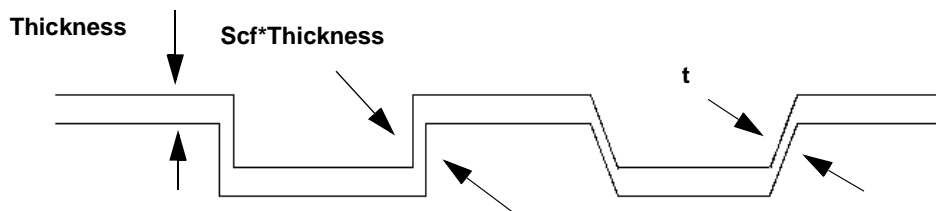
**Deposit** types include **CONFORMAL**, **SNOWFALL**, and **FILL**:

- **CONFORMAL** deposit adds a layer that follows the contour of the processed wafer.
- **SNOWFALL** covers only those surfaces that are not shadowed by other surfaces on the wafer.
- **FILL** is a maskless **Process Step** that makes the surface of the wafer plane.

Each **DepositType** has unique parameter requirements.



*DepositType = CONFORMAL*

A **CONFORMAL** deposit is illustrated below.



Parameters for **CONFORMAL** deposits are described below.

<i>Parameter</i>	<i>Description</i>
<b>WaferID</b>	This optional parameter identifies the <b>Wafer</b> that will receive the deposit. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.



<i>Parameter</i>	<i>Description</i>
<b>DepositType</b>	Type of deposit. Parameter options are <b>CONFORMAL</b> , <b>SNOWFALL</b> , and <b>FILL</b> . The value is set to <b>CONFORMAL</b> for a conformal deposit.
<b>Face</b>	Side of the <b>Wafer</b> that will receive the deposit. Parameter options are <b>TOP</b> , <b>BOT</b> , and <b>TOPBOT</b> , for both top and bottom. Do not enclose in quotation marks.
<b>LayerName</b>	Name of the layer to be deposited. Any valid L-Edit layer name enclosed in quotation marks is an acceptable entry for <b>LayerName</b> . <b>LayerName</b> and <b>Target</b> are typically set to the same value.
<b>Thickness</b>	You may enter any positive value for the vertical dimension of the <b>CONFORMAL</b> deposit. This thickness is deposited on the sides(s) specified by the <b>Face</b> parameter.

<i>Parameter</i>	<i>Description</i>
<b>Scf</b>	<p>The <b>Scf</b> (<i>Sidewall coverage factor</i>) parameter is the height of the material deposited on vertical sidewalls divided by the <b>Thickness</b> of the material deposited on horizontal surfaces of a <b>CONFORMAL</b> deposit. Material coverage (<b>t</b>) on intermediate slopes depends on the angle of the wall according to the relationship described in the section on Thickness and Scf on page 639. Entries for <b>Scf</b> can be a decimal number between <b>0</b> and <b>1</b>, or the letter <b>c</b>, and, whether numeral or character, must be enclosed in quotation marks. An <b>Scf</b> of <b>c</b> is equivalent to an <b>Scf</b> of <b>1.0</b>, which is a completely conformal deposit, that is, a deposit with uniform thickness along the wafer contour. (The <b>Scf</b> parameter is not supported in MEMS Pro Version 3 and therefore not required. The value of this parameter is assumed to be <b>1.0</b> or <b>c</b> for this release.).</p>
<b>Target</b>	<p>This parameter specifies the 3D model rendering characteristics of the deposited layer. You may enter any valid L-Edit layer name as text enclosed in quotation marks for <b>Target</b>. <b>Target</b> and <b>LayerName</b> are typically set to the same value. For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 114.</p>
<b>Label</b>	<p>During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.</p>

<i>Parameter</i>	<i>Description</i>
------------------	--------------------

<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.
----------------	---

### *Thickness and Scf*

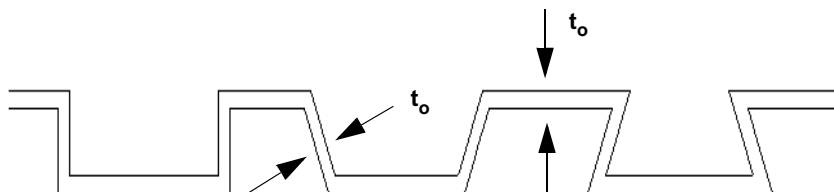
The relationship between **Thickness** and **Scf** (*Sidewall coverage factor*) can be clarified using some diagrams.

Assume that the profile below has been created by previous steps in the fabrication process.



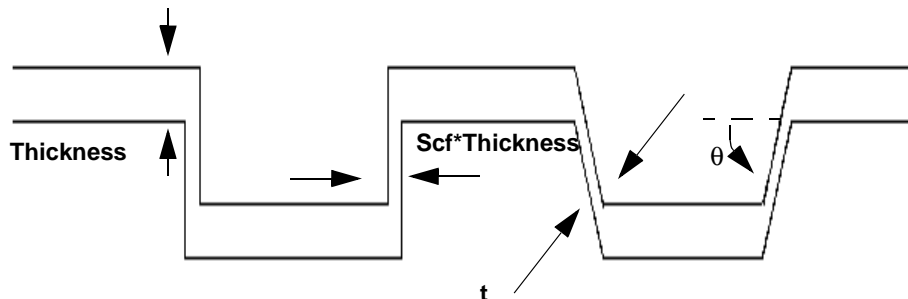


Setting **DepositType** = **CONFORMAL**, **Scf** = **1.0 = c**, and **Thickness** =  **$t_o$**  (for example, MUMPS deposition of ***Poly1***), the profile becomes the following:



Material has been evenly deposited across the entire surface of the wafer.

If, however, **Scf** is set to a value between 0 and 1, the coverage will depend on the existing geometry, as the following diagram illuminates.



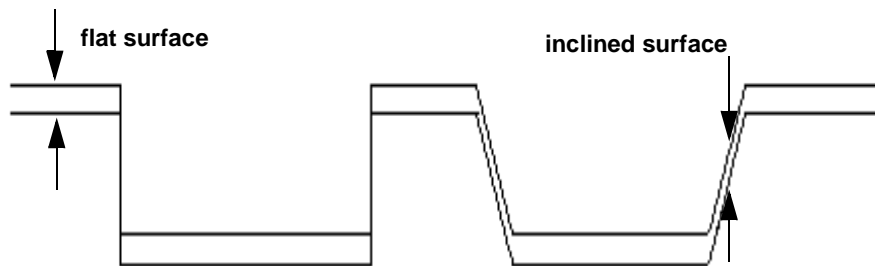
The thickness **t** on sidewalls at intermediate angles depends on the angle of inclination  $\theta$  of the sidewall and **CONFORMAL** parameters **Scf** and **Thickness** according to the following expression:

$$t = \text{Thickness} \sqrt{\cos^2(\theta) + \text{Scf}^2 \sin^2(\theta)}$$

When the angle  $\theta$  is  $90^\circ$ , the expression for **t** reduces to **Scf\*Thickness**, as it must, by the definition of **Scf**. Setting **Scf** to 0.5 deposits half as much material on the sides of a  $90^\circ$  wall as on the top.

## *DepositType = SNOWFALL*

**SNOWFALL** deposits no material on vertical and shadowed surfaces, as shown below. Horizontal surfaces have the deepest coverage. Inclined surfaces have an intermediate amount of material deposited upon them.



Parameters for **SNOWFALL** deposits are described below.

<i>Parameter</i>	<i>Description</i>
<b>WaferID</b>	This optional parameter identifies the <b>Wafer</b> that will receive the deposit. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.

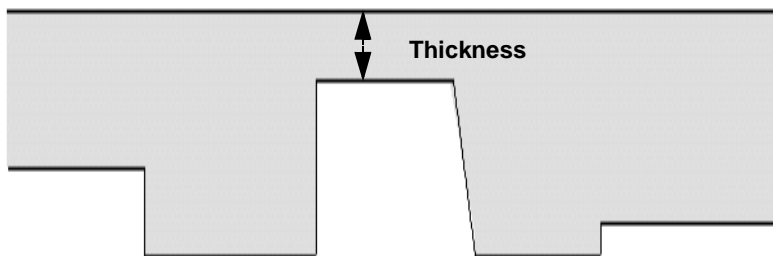
<i>Parameter</i>	<i>Description</i>
<b>DepositType</b>	Type of deposit. Parameter options are <b>CONFORMAL</b> , <b>SNOWFALL</b> , and <b>FILL</b> . The value is set to <b>SNOWFALL</b> for a snowfall deposit.
<b>Face</b>	Side of the <b>Wafer</b> to receive the deposit. Parameter options are <b>TOP</b> , <b>BOT</b> , and <b>TOPBOT</b> , for both top and bottom. Do not enclose in quotation marks.
<b>LayerName</b>	Name of the layer to be deposited. Any valid L-Edit layer name enclosed in quotation marks is an acceptable entry for <b>LayerName</b> . <b>LayerName</b> is often set to the same value as <b>Target</b> .
<b>Target</b>	<b>Target</b> specifies the 3D model rendering characteristics of the deposited layer. You may enter any valid L-Edit layer name as text enclosed in quotation marks for <b>Target</b> . <b>Target</b> is often set to the same value as <b>Layername</b> . For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 114.
<b>Thickness</b>	Any positive decimal number may be entered for the vertical dimension of the <b>SNOWFALL</b> deposit. This thickness is deposited on the side(s) specified by the <b>Face</b> parameter.
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.

<i>Parameter</i>	<i>Description</i>
------------------	--------------------

<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.
----------------	---

*DepositType = FILL*

As illustrated below, the **Thickness** of **FILL** is set from the highest point on the model at that step for the **TOP Face**.



Parameters for **FILL** deposits are described below.

<i>Parameter</i>	<i>Description</i>
<b>WaferID</b>	Identifies the <b>Wafer</b> to be filled. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.
<b>DepositType</b>	Type of deposit. Parameter options are <b>CONFORMAL</b> , <b>SNOWFALL</b> , and <b>FILL</b> . The value is set to <b>FILL</b> for a fill deposit.
<b>Face</b>	Side of the <b>Wafer</b> to be filled. Parameter options are <b>TOP</b> , <b>BOT</b> , and <b>TOPBOT</b> , for both top and bottom. Do not enclose in quotation marks.
<b>LayerName</b>	Name of the layer to be deposited. Any valid L-Edit layer name enclosed in quotation marks is an acceptable entry for <b>LayerName</b> . <b>Layername</b> is often set to the same value as <b>Target</b> .

<i>Parameter</i>	<i>Description</i>
<b>Target</b>	This parameter specifies the 3D model rendering characteristics of the filled layer. You may enter any valid L-Edit layer name as text enclosed in quotation marks for <b>Target</b> . <b>Target</b> is often set to the same value as <b>LayerName</b> . For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 114.
<b>Thickness</b>	The vertical dimension of the <b>FILL</b> deposit as measured from the highest point on the <b>Wafer</b> up for the <b>TOP</b> face, or from the lowest point of the <b>Wafer</b> down for the <b>BOT</b> face (See the figure on page 377). <b>Thickness</b> may be any positive decimal number. The material is deposited on the side(s) specified by the <b>Face</b> parameter.
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.
<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.

## Etch

The syntax for the **Etch** command is presented below, followed by an example containing valid entries for its parameters.

### *Syntax*

```
Etch={  
    WaferID=S  
    EtchType=etype  
    Face=face  
    MaskName=layer  
    EtchMask=emask  
    Depth=P  
    Angle=A  
    Undercut=R  
    EtchRemoves=layer  
    EtchRemoves=layer  
    Label=S  
    Comment=S  
}
```

### *Example*

```
Etch={  
    EtchType=SURFACE  
    Face=TOP
```



```
MaskName = "Anchor1"  
EtchMask=INSIDE  
Depth = 2.5  
Angle = 87  
Undercut = 0  
EtchRemoves = "ox1"  
WaferID="w1"  
Label = "Etch Anchor1"  
Comment = "Etch Anchor1"  
}
```

### *Description*

There are three types of etches: **SURFACE**, **BULK**, and **SACRIFICIAL**:

- **SURFACE** etches remove material that has been deposited during previous steps.
- **BULK** etches remove parts of the **Wafer**.
- **SACRIFICIAL** etches completely remove all bodies on the **EtchRemoves** layers. This etch does not require masking, and therefore has no setting for the **EtchMask** or **MaskName** parameter.

The orientation of the **Wafer** must be taken into account when setting these parameters.

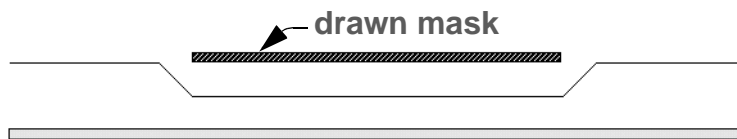
## Orientation Considerations

The **Face** to be etched may be the top face (**TOP**), the bottom face (**BOT**), or both faces simultaneously (**TOPBOT**). If you are designing masks for processing on both faces of the wafer, you must be careful of the orientation of the masks. As Alan Nutt of Kodak Research Laboratories points out, to ensure correct alignment (as drawn in layout) of the masks designed for processing on the bottom of the wafer with the masks designed for processing on the top of the wafer, the former must be flipped horizontally (i.e., left-right reversed). You may be required to perform the reversal yourself or have the mask maker perform it. Please consult your mask maker for further information.

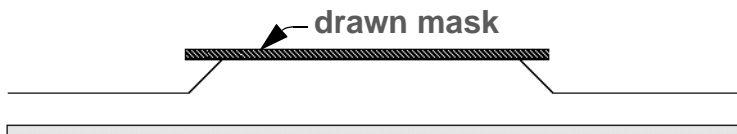
Another consideration for **SURFACE** etch is whether the mask setting is *inclusive* or *exclusive*. **EtchMask** may be set to either **INSIDE** (inclusive) or **OUTSIDE** (exclusive).

For **EtchMask = INSIDE**, areas beneath the mask layer geometry are removed (generally used for insulator masks). For **EtchMask = OUTSIDE**, areas beneath the mask layer geometry are protected (generally used for conductor masks). Below, identical masks with differing **EtchMask** settings illustrate this effect:

**EtchMask = INSIDE**



**EtchMask = OUTSIDE**



**SURFACE, BULK, and SACRIFICIAL** etches are described below.

## *EtchType = SURFACE*

The outcome of a **SURFACE** etch depends on **EtchMask**, **Angle**, **Undercut**, and **Depth**. In the diagram below, the **EtchMask** parameter is set to **OUTSIDE**.



Parameters for a **SURFACE** etch are described below.

<i>Parameter</i>	<i>Description</i>
<b>WaferID</b>	This optional parameter identifies the <b>Wafer</b> to be etched. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.

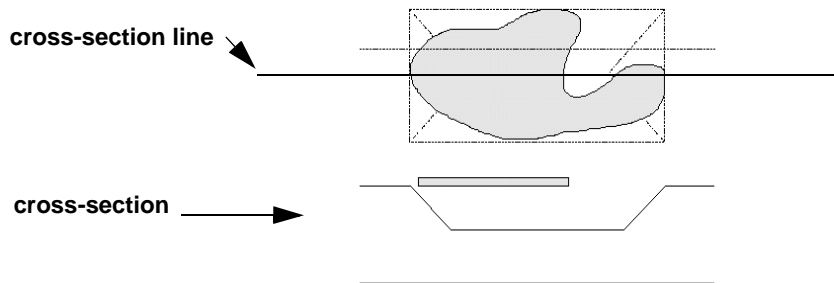
<i>Parameter</i>	<i>Description</i>
<b>EtchType</b>	Type of etch. Parameter options are <b>SURFACE</b> , <b>BULK</b> , and <b>SACRIFICIAL</b> . The value is set to <b>SURFACE</b> for a surface etch.
<b>Face</b>	Side of the <b>Wafer</b> to be etched. Parameter options are <b>TOP</b> , <b>BOT</b> , and <b>TOPBOT</b> , for both top and bottom. Do not enclose in quotation marks.
<b>MaskName</b>	Any valid L-Edit layer name enclosed in quotation marks is an acceptable input for <b>MaskName</b> . The geometry on this mask defines the area to be etched or excluded from etching.
<b>EtchMask</b>	Parameter options are <b>INSIDE</b> or <b>OUTSIDE</b> . This parameter sets the mask orientation, that is whether the material to be removed is <b>INSIDE</b> or <b>OUTSIDE</b> of the drawn layout. Do not enclose in quotation marks.
<b>Depth</b>	Vertical dimension of the etch. Only the layers specified in the <b>EtchRemoves</b> parameter will be affected by the etch. If the <b>Depth</b> is greater than the <b>Thickness</b> of the etched layer, the layer beneath will not be removed. <b>Depth</b> may be any positive decimal number.

<i>Parameter</i>	<i>Description</i>
<b>Angle</b>	The etch <b>Angle</b> is determined in decimal degrees between 45 and 90. <b>Angle</b> is not supported for <b>SURFACE</b> etch under MEMS Pro Version 3 and therefore not required. In the current version of the program, the 3D Modeler assumes that the etch <b>Angle</b> is 90° for <b>SURFACE</b> etches.
<b>Undercut</b>	<p>For <b>EtchMask = INSIDE</b>, <b>Undercut</b> is the distance the etch front will extend over the drawn mask edge. For <b>EtchMask = OUTSIDE</b>, <b>Undercut</b> is the distance the etch front will intrude under the drawn mask edge. <b>Undercut = 0</b> is a sharply defined cut, aligned to the mask edge for both cases. <b>Undercut</b> may be any positive decimal number.</p> <p><b>Undercut</b> is not supported under MEMS Pro Version 3 and therefore not required. In the current version of the program, the 3D Modeler assumes that there will be no <b>Undercut</b>, i.e., <b>Undercut = 0</b> for <b>SURFACE</b> etches.</p>
<b>EtchRemoves</b>	Name of the layer that will be removed by this etch step. Any valid L-Edit layer name enclosed in quotation marks is an acceptable entry for <b>EtchRemoves</b> . There may be multiple entries of this parameter.
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.

<i>Parameter</i>	<i>Description</i>
<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.

*EtchType* = **BULK**

The **BULK** etch sketched below is of KOH or EDP on a silicon wafer of **100** crystal orientation. The pit is bound by the **111** plane, which is attacked at a much slower rate than all other crystallographic planes. The outline of the box is the minimum bounding box of the mask pattern. This etch assumes **EtchMask** = **INSIDE**. The etch is viewed from above the **TOP** face. A cross-section corresponding to the dashed line appears below.



Parameters for the **BULK** etch are described below.

Recall that the **BULK** etch is designed to remove the **Wafer** material only. There is no need to identify the **Wafer** with the **EtchRemoves** parameter.

<i>Parameter</i>	<i>Description</i>
<b>WaferID</b>	This optional parameter identifies the <b>Wafer</b> to be etched. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are any text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.
<b>EtchType</b>	Type of etch. Parameter options are <b>SURFACE</b> , <b>BULK</b> , and <b>SACRIFICIAL</b> . The value is set to <b>BULK</b> for a bulk etch.
<b>Face</b>	Side of the <b>Wafer</b> to be etched. Parameter options are <b>TOP</b> , <b>BOT</b> , and <b>TOPBOT</b> , for both top and bottom. Do not enclose in quotation marks.
<b>MaskName</b>	Any valid L-Edit layer name enclosed in quotation marks is an acceptable input for <b>MaskName</b> . The geometry on this mask defines the area to be etched or excluded from etching.
<b>Depth</b>	Vertical dimension of the etch. Any positive decimal number may be entered for <b>Depth</b> .
<b>Angle</b>	Etch <b>Angle</b> is given in decimal degrees between 45 and 90.



<i>Parameter</i>	<i>Description</i>
<b>Undercut</b>	The distance the etching material will extend over the mask edge. <b>Undercut</b> may be any positive decimal number. <b>Undercut = 0</b> is a sharply defined cut, aligned to the mask edge.
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.
<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.

Note that **Angle** and **Undercut** are supported for **Bulk** etches in MEMS Pro Version 3.

## *EtchType = SACRIFICIAL*

Parameters for a **SACRIFICIAL** etch are illustrated below.

<i>Parameter</i>	<i>Description</i>
<b>WaferID</b>	This optional parameter identifies the <b>Wafer</b> to be etched. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.
<b>EtchType</b>	Type of etch. Parameter options are <b>SURFACE</b> , <b>BULK</b> , and <b>SACRIFICIAL</b> . The value is set to <b>SACRIFICIAL</b> for a sacrificial etch.
<b>Face</b>	Side of the <b>Wafer</b> to be etched. Parameter options are <b>TOP</b> , <b>BOT</b> , and <b>TOPBOT</b> , for both top and bottom. Do not enclose in quotation marks.
<b>EtchRemoves</b>	Name of layer to be removed by this etch. Any valid L-Edit layer name enclosed in quotation marks is an acceptable entry for <b>EtchRemoves</b> . There may be multiple entries of this parameter.

<i>Parameter</i>	<i>Description</i>
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.
<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.

## MechanicalPolish

The syntax for the **MechanicalPolish** command is given below, followed by an example containing valid entries for its parameters.

### *Syntax*

```
MechanicalPolish={  
    WaferID=S  
    Face=face  
    Depth=P  
    Thickness=P  
    Label=S  
    Comment=S  
}
```

### *Example*

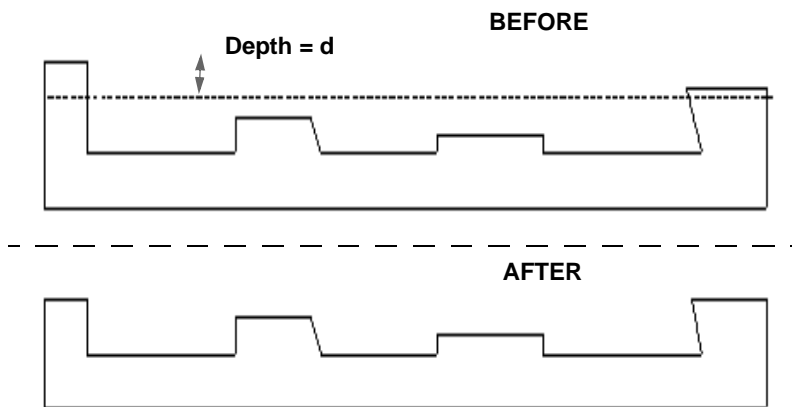
```
MechanicalPolish={  
    WaferID="w1"  
    Face=TOP  
    Depth=23.0  
    Label="Mechanical Polish"  
    Comment="Mechanical Polish"  
}
```

## Description

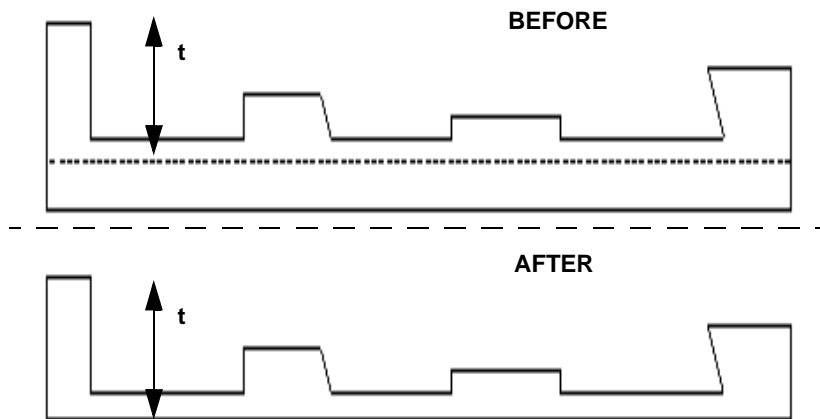
**MechanicalPolish** truncates the specified **Depth** off the top or bottom of the entire wafer regardless of material type.

The effects of **MechanicalPolish** can be specified by either a **Depth** or a **Thickness**, *but not both*. When a **Depth** is specified, that **Depth** is truncated off the face of the wafer. When a **Thickness** is specified, that **Thickness** remains after polishing.

The drawing below gives the profile of a wafer before and after **MechanicalPolish**. The depth **d** has been sliced off the top of the wafer.



In the drawing below, the **MechanicalPolish** command has sliced material from the bottom of the **Wafer** and left **Thickness** = **t**.



Parameters for **MechanicalPolish** are described below.

<i>Parameter</i>	<i>Description</i>
<b>WaferID</b>	This optional parameter identifies the <b>Wafer</b> to be polished. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.
<b>Face</b>	Side of the <b>Wafer</b> to be polished. Parameter options are <b>TOP</b> or <b>BOT</b> . Only one face may be polished at a time in Version 3. Do not enclose in quotation marks.
<b>Depth</b>	Vertical measure of the material <i>removed</i> , measured from the highest point of the <b>Wafer</b> for the <b>TOP</b> face, or from the lowest point of the <b>Wafer</b> for the <b>BOT</b> face. <b>Depth</b> may be any positive decimal number.
<b>Thickness</b>	Vertical measure of the material that <i>remains</i> after the polish. It is measured from the lowest point of the <b>Wafer</b> for the <b>TOP</b> face and from the highest point of the <b>Wafer</b> for the <b>BOT</b> face. Any positive decimal number may be entered for <b>Thickness</b> .
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.

---

<b>WaferID</b>	This optional parameter identifies the <b>Wafer</b> to be polished. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.
<b>Face</b>	Side of the <b>Wafer</b> to be polished. Parameter options are <b>TOP</b> or <b>BOT</b> . Only one face may be polished at a time in Version 3. Do not enclose in quotation marks.
<b>Depth</b>	Vertical measure of the material <i>removed</i> , measured from the highest point of the <b>Wafer</b> for the <b>TOP</b> face, or from the lowest point of the <b>Wafer</b> for the <b>BOT</b> face. <b>Depth</b> may be any positive decimal number.
<b>Thickness</b>	Vertical measure of the material that <i>remains</i> after the polish. It is measured from the lowest point of the <b>Wafer</b> for the <b>TOP</b> face and from the highest point of the <b>Wafer</b> for the <b>BOT</b> face. Any positive decimal number may be entered for <b>Thickness</b> .
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.

<i>Parameter</i>	<i>Description</i>
------------------	--------------------

<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.
----------------	---



## ImplantDiffuse

**ImplantDiffuse** is not supported in MEMS Pro Version 3.

The syntax for the **ImplantDiffuse** command is given below, followed by an example containing valid entries for its parameters.

### *Syntax*

```
ImplantDiffuse={  
    WaferID=S  
    Face=face  
    MaskName=layer  
    Depth=P  
    Angle=A  
    Undercut=P  
    Target=layer  
    Source=layer  
    Label=S  
    Comment=S  
}
```

### *Example*

```
ImplantDiffuse={  
    WaferID="w1"  
    Face=TOP
```

```
MaskName="NOxide 1"  
Depth=4  
Angle=90  
Undercut=0  
Target= "Poly1"  
Source="NOxide"  
Label= "ImplantDiffuse"  
Comment="Nwell Source for Implant"  
}
```

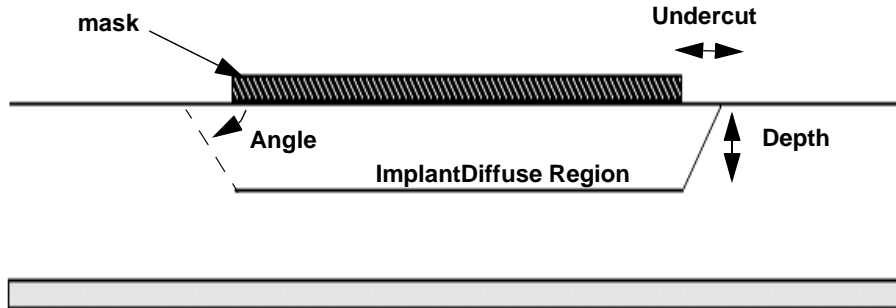
### *Description*

This step models the implantation and diffusion of impurities into a material. The result of this step is similar to the **EtchType = SURFACE**, **EtchMask = INSIDE** step except that the etched part is replaced, not removed. The **Depth**, **Undercut**, and **Angle** parameters model the geometric effects of the **ImplantDiffuse** step. The **Depth** parameter represents the vertical junction depth. The **Undercut** and **Angle** parameters model the lateral diffusion effect. The **Undercut** sets the extent to which the impurities spread over the edge of the mask. Notice that the actual mask used during fabrication is the boolean negative of the drawn mask (shown in the diagram). The **Angle** parameter models the curved profile of the diffusion region as a straight line.

## Orientation Considerations

The **Face** to be etched may be the top face (**TOP**), the bottom face (**BOT**), or both faces simultaneously (**TOPBOT**). If you are designing masks for processing on both faces of the wafer, you must be careful of the orientation of the masks. As Alan Nutt of Kodak Research Laboratories points out, to ensure correct alignment (as drawn in layout) of the masks designed for processing on the bottom of the wafer with the masks designed for processing on the top of the wafer, the former must be flipped horizontally (i.e., left-right reversed). You may be required to perform the reversal yourself or have the mask maker perform it. Please consult your mask maker for further information.

Some parameters for **ImplantDiffuse** are illustrated below.



Parameters for **ImplantDiffuse** are described below.

<i>Parameter</i>	<i>Description</i>
<b>WaferID</b>	This optional parameter identifies the <b>Wafer</b> to be implanted/diffused. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries are text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.
<b>Face</b>	Side of the <b>Wafer</b> to be implanted /diffused. Parameter options are <b>TOP</b> , <b>BOT</b> , and <b>TOPBOT</b> , for both top and bottom. Do not enclose in quotation marks.
<b>MaskName</b>	Name of the <i>inclusive</i> mask layer. The areas drawn on this layer will be affected by <b>ImplantDiffuse</b> . Any valid L-Edit layer name enclosed in quotation marks is an acceptable entry for <b>MaskName</b> .
<b>Depth</b>	Vertical measure of the extent of <b>ImplantDiffuse</b> . Any positive decimal number is an acceptable entry for <b>Depth</b> .
<b>Angle</b>	<b>Angle</b> in decimal degrees. Any decimal number between 45 and 90 is an acceptable entry for <b>Angle</b> .

<i>Parameter</i>	<i>Description</i>
<b>Undercut</b>	The distance the implanted/diffused material will extend outward from the mask edge. <b>Undercut = 0</b> is a sharply defined cut, aligned to the mask edge. <b>Undercut</b> may be any positive decimal number.
<b>Target</b>	The name of the layer that will receive the implanted/ diffused material. Any valid L-Edit layer name enclosed in quotation marks may be entered for <b>Target</b> . There may be multiple entries of this parameter.
<b>Source</b>	Layer name of the material that is being introduced into the <b>Target</b> . 3D rendering information for the <b>ImplantDiffuse</b> step is derived from this layer. Any valid L-Edit layer name enclosed in quotation marks is an acceptable entry for <b>Source</b> . For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 114.
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.
<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.

Note that only **Target** is affected by **ImplantDiffuse**. If the targeted **Depth** is greater than the **Target's Thickness**, the layer beneath is not affected.

## Grow

**Grow** is not supported in MEMS Pro Version 3.

The syntax for the **Grow** command is presented below, followed by an example containing valid entries for its parameters.

### *Syntax*

```
Grow={  
    WaferID=S  
    Face=face  
    MaskName=layer  
    Thickness=P  
    Depth=percent  
    Undercut=P  
    Target=layer  
    Source=layer  
    Label=S  
    Comment=S  
}
```

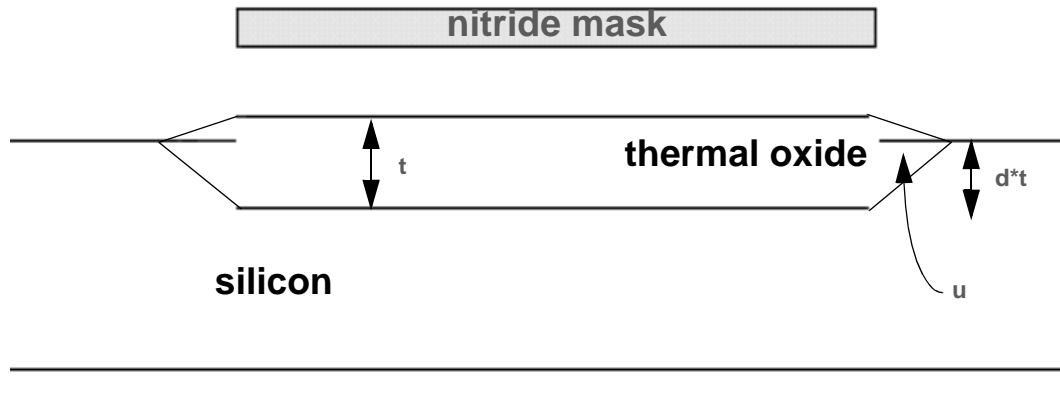
### *Example*

```
Grow={  
    WaferID="w1"  
    Face=TOP
```

```
MaskName="Grow"  
Thickness=7.0  
Depth=40  
Undercut=5  
Target="Silicon 1"  
Source="Silicon Dioxide"  
Label="Grow"  
Comment="Grow"  
}
```

### *Description*

**Grow** consolidates the processes of nitride deposition, nitride etch, oxide growth, and nitride removal into a single command. A typical **Grow** step might be localized oxidation of silicon (LOCOS). The **Depth**, **Thickness**, and **Undercut** parameters model the geometric effects of the **Grow** step. The effect of this step with **Depth = d**, **Thickness = t**, and **Undercut = u** is shown in the following diagram.





Parameters for **Grow** are described below.

<i>Parameters</i>	<i>Description</i>
<b>WaferID</b>	This optional parameter identifies the <b>Wafer</b> to be grown. Since MEMS Pro Version 3 supports just one <b>Wafer</b> , the <b>WaferID</b> is assigned automatically and is read-only. The default value of <b>w1</b> will override any entry made for <b>WaferID</b> , where valid entries will be text enclosed in quotation marks. Future versions of the software will support multiple wafers and user-assigned names.
<b>Face</b>	Side of the <b>Wafer</b> to be grown. Parameter options are <b>TOP</b> , <b>BOT</b> , and <b>TOPBOT</b> , for both top and bottom. Do not enclose in quotation marks.
<b>MaskName</b>	Name of the <i>inclusive</i> mask layer. The areas drawn on this layer will be affected by <b>Grow</b> . Any valid L-Edit layer name enclosed in quotation marks is an acceptable entry for <b>MaskName</b> .
<b>Thickness</b>	Any positive decimal number may be entered for the vertical height of the growth.

<i>Parameters</i>	<i>Description</i>
<b>Depth</b>	Maximum intrusion of the source material into the target as a percentage of the total <b>Thickness</b> . <b>Depth = 50</b> implies that growth embeds itself (or, alternatively, expands) within the <b>Target</b> to half of the total <b>Thickness</b> . <b>Depth</b> must be a positive decimal number between 0 and 100.
<b>Undercut</b>	The distance the growth will extrude outward from the mask edge. <b>Undercut = 0</b> is aligned to the mask edge. Any positive decimal number may be entered for <b>Undercut</b> .
<b>Target</b>	The <b>Target</b> is the material that is reduced or consumed by the chemical process that produces the <b>Source</b> . Any valid L-Edit layer name may be entered as text enclosed in quotation marks to identify the <b>Target</b> . There may be multiple entries for <b>Target</b> .
<b>Source</b>	Layer name of the material constructed on the <b>Target</b> . It is the primary reference for 3D rendering information for the <b>Grow</b> step. For example, silicon dioxide is the <b>Source</b> for thermal oxide growth on exposed silicon.  Any valid L-Edit layer name enclosed in quotation marks is an acceptable entry for <b>Source</b> . For more information on 3D model rendering characteristics, see Defining Colors for 3D Models on page 114.

<i>Parameters</i>	<i>Description</i>
<b>Label</b>	During 3D model generation, this string, describing the ongoing step, will be displayed in the progress dialog. The <b>Label</b> may be any text enclosed in quotation marks.
<b>Comment</b>	<b>Comment</b> may be any text enclosed in quotation marks.

## Editing the Process Definition

You may graphically edit the process definition. To do this, access the **Process Definition** dialog by selecting **3D Tools > Edit Process Definition** in the MEMS Pro Palette. For detailed instructions, see [Editing the Process Definition](#) on page 149.

## Process Definition Example: MUMPs

The following process definition file describes the MCNC MUMPs 3 polysilicon layer process. The form of a particular 3D model created using this process definition depends on its specific mask layouts.

```
ProcessInfo = {  
    Name = "MUMPS"  
    Version = "1.0beta"  
    Unit = microns  
}  
Wafer={  
    MaskName="substrate"  
    Thickness=5  
    WaferID="w1"  
    Target="substrate"  
    Label="Wafer"  
    Comment="Wafer"  
}  
Deposit={  
    DepositType=CONFORMAL  
    Face=TOP  
    LayerName="nitride"  
    Thickness=.6  
    Scf="c"  
    WaferID="w1"  
    Target="nitride"
```

```
        Label = "Deposit Nitride"
        Comment = "Deposit Nitride"
    }
    Deposit={
        DepositType=CONFORMAL
        Face=TOP
        LayerName="Poly0"
        Thickness=.5
        Scf="c"
        WaferID="w1"
        Target="Poly0"
        Label="Deposit Poly0"
        Comment="Deposit Poly0"
    }
    Deposit={
        DepositType=CONFORMAL
        Face=TOP
        LayerName="ox1"
        Thickness=2
        Scf=".5"
        WaferID="w1"
        Target="ox1"
        Label="Deposit Ox1"
        Comment="Deposit Ox1"
    }
    Etch={
        EtchType=SURFACE
        Face = TOP
        MaskName = "Dimple"
```

```
EtchMask=INSIDE
Depth = .75
Angle = 87
Undercut = 0
EtchRemoves = "ox1"
WaferID="w1"
Label = "Etch Dimple"
Comment = "Etch Dimple"
}
Deposit = {
    DepositType=CONFORMAL
    Face=TOP
    LayerName="Poly1"
    Thickness=2
    Scf="c"
    WaferID="w1"
    Target="Poly1"
    Label="Deposit Poly1"
    Comment="Deposit Poly1"
}
Etch={
    EtchType=SURFACE
    Face=TOP
    MaskName="Poly1"
    EtchMask=OUTSIDE
    Depth=4.5
    Angle=90
    Undercut=0
    EtchRemoves="Poly1"
```

```
        WaferID="w1"
        Label = "Etch Poly1"
        Comment = "Etch Poly1"
    }
    Deposit = {
        DepositType=CONFORMAL
        Face=TOP
        LayerName="ox2"
        Thickness=.75
        Scf = ".5"
        WaferID="w1"
        Target="ox2"
        Label = "Deposit Ox2"
        Comment = "Deposit Ox2"
    }
    Etch = {
        EtchType=SURFACE
        Face=TOP
        MaskName = "Poly1-Poly2 Via"
        EtchMask = INSIDE
        Depth = 1.5
        Angle = 87
        Undercut = 0
        EtchRemoves = "ox2"
        EtchRemoves = "ox1"
        WaferID="w1"
        Label = "Etch Poly1-Poly2 Via"
        Comment = "Etch Poly1-Poly2 Via"
    }
```



```
Etch= {
    EtchType=SURFACE
    Face=TOP
    MaskName = "Anchor2"
    EtchMask=INSIDE
    Depth = 5.25
    Angle = 87
    Undercut = 0
    EtchRemoves = "ox2"
    EtchRemoves = "ox1"
    WaferID="w1"
    Label ="Etch Anchor2"
    Comment = "Etch Anchor2"
}
Deposit = {
    DepositType=CONFORMAL
    Face = TOP
    LayerName = "Poly2"
    Thickness = 1.5
    Scf="c"
    WaferID="w1"
    Target = "Poly2"
    Label = "Deposit Poly2"
    Comment = "Deposit Poly2"
}
Etch = {
    EtchType=SURFACE
    Face = TOP
    MaskName = "Poly2"
```

```
EtchMask=OUTSIDE
Depth = 6.75
Angle = 90
Undercut = 0
EtchRemoves = "Poly2"
WaferID="w1"
Label = "Etch Poly2"
Comment = "Etch Poly2"
}
Deposit = {
    DepositType=SNOWFALL
    Face =TOP
    LayerName = "Metal"
    Thickness = .52
    WaferID="w1"
    Target = "Metal"
    Label = "Sputter Metal"
    Comment = "Sputter Metal"
}
Etch= {
    EtchType=SURFACE
    Face =TOP
    MaskName = "Metal"
    EtchMask=OUTSIDE
    Depth = .52
    Angle = 90
    Undercut = 0
    EtchRemoves = "Metal"
    WaferID="w1"
```

```
        Label = "Metal Liftoff"  
        Comment = "Metal Liftoff"  
    }  
    Etch = {  
        EtchType=SACRIFICIAL  
        Face=TOP  
        EtchRemoves = "ox1"  
        EtchRemoves = "ox2"  
        WaferID="w1"  
        Label = "Sacrificial Etch"  
        Comment = "Sacrificial Etch"  
    }
```

# INDEX

---

## Numerics

### 3D model

- deletion, 119, 178
- edition, 353
- export, 120, 180, 378
- generation, 77
- importing to ANSYS, 380
- meshing in ANSYS, 388
- view, 72, 81, 118, 149
- viewing in ANSYS, 380

### 3D Model View toolbar, 169

### 3D Modeler

- accelerometer, 138
- cross-section, 175
- defining colors for models, 147

deleting a model, 178  
diaphragm, 141  
error checking, 212  
exporting 3D models, 180  
input formats, 143  
menu bar  
    File menu, 153  
    Help menu, 167  
    Setup menu, 165  
    Tools menu, 164  
    View menu, 156  
    Window menu, 166  
menu bar, 153  
multiple views, 79  
output formats, 143  
palette, 170  
rotary motor, 136  
status bar, 173  
thermal actuator, 134  
title bar, 152  
toolbar, 169  
user interface, 151  
viewing 3D models, 149

3D Modeler, 7  
3D To Layout  
    Add Volumes, 310  
    Create Volumes, 305  
    Delete Volumes, 309  
    Export CIF File, 319  
    Import MEMS, 302, 345  
    Save MEMS, 317  
3D To Layout menu, 298  
3D Tools  
    Delete 3D Model, 178  
    Edit Process Definition, 185, 663  
3D Tools menu  
    Delete 3D Model, 119  
    Edit Process Definition, 116  
    Export 3D Model, 120  
    View 3D Model, 118  
3D Tools menu, 116, 145

## A

Accelerometer, 138

Add Volumes, 310

Adding

    volumes, 310, 356

Alignment macro, 577

All angle wire, 88

Analog Devices/MCNC iMEMS, 607

Analysis

    running, 391

    viewing the results, 392

ANF file format, 144

ANSYS

    accessing the R.O.M menu, 218

    adding an element type, 383

    ANSYS Neutral Format (.anf), 144

    connection product for SAT 144, 182, 378, 379

    importing 3D models, 380

    linear structural analysis, 391

    links to, 8

    meshing a model, 388

    running an analysis, 391

    setting boundary conditions, 384

    setting material properties, 382

    viewing a 3D model, 380

viewing the results, 392

ANSYS to layout

- editing a 3D model, 353
- limitations, 339

ANSYS to Layout dialog box, 323

ANSYS toolbar

- LAYOUT, 320

Approximation, 420

Area-perimeter dielectric isolation test structure, 538

Array

- Euler columns, 548
- Guckel ring, 554

## B

Beam

- doubly supported (Euler column), 545
- linear folded, 532

Bidirectional rotary comb-drive, 508

Block place and route

- Voir BPR

Bonding Pad



generation, 63  
Bonding pad, 568  
Boundary conditions  
    setting in ANSYS, 384  
Box  
    drawing, 94  
BPR  
    initialization, 432  
    routing a design, 443  
BPR, 7, 430  
BULK etch type, 202, 206

## C

Checking  
    errors in 3D model, 212  
CIF file  
    export, 319  
Circle  
    drawing, 94  
Clearing Vertex Information, 130  
Comb-drive

bidirectional rotary, 508  
generation, 59  
instantiation, 22  
linear electrostatic, 496  
rotary, 511  
unidirectional rotary (1), 502  
unidirectional rotary (2), 505  
Comb-drive, 561  
Command tool  
    accessing, 400  
    resulting schematic object, 405  
Command Tool dialog 402  
Command tool, 398  
Components  
    definition, 310  
Condensation algorithm, 222  
CONFORMAL deposit, 195, 624  
Connecting  
    global nodes, 27  
Crab-leg  
    linear (1), 526  
    linear (2), 529  
Create Property dialog box, 407

Create Spline dialog box, 412

Create Volumes, 305

Creating

- a hole, 121

- a module, 20

- a new property, 407

- a schematic symbol, 472

- a schematic, 17

- splines, 125, 415

- volumes, 305, 353

Crossover test structure

- type 1, 541

- type 2, 543

Cross-section

- view, 175

Cross-section, 81

## D

Defining

- components, 310

Delete 3D Model, 119

Delete Volumes, 309

Deleting

- a 3D model, 119, 178

- volumes, 309, 360

Deposit type

- CONFORMAL, 195, 624

- FILL, 199, 632

- SNOWFALL, 197, 630

Deposit, 194, 622

Design

- optimization, 450

Diaphragm, 141

Drawing

- a box, 94

- a circle, 94

- a curved polygon, 90

- a spline, 95

- a torus, 88

- a wire, 87

Drawing, 85

Dual archimedean spiral spring, 535

## E

Easy MEMS menu

- Plate Release, 121

- Polar Array, 123

Easy MEMS menu, 121

Edit component, 112

Edit Process Definition, 116, 663

Editing

- a 3D model, 353

- a process definition, 116, 185

- a process, 663

- a spline, 127

- generated layout parameters, 494

- object properties, 30

- process steps parameters, 190

- process steps, 186, 188

- splines, 429

Element type

- addition in ANSYS, 383

Etch type

- BULK, 202, 206

- SACRIFICIAL, 202, 208

SURFACE, 202  
SURFACE,SURFACE etch type, 203  
Etch, 201, 635  
Euler column  
    array, 548  
Euler Column, 545  
Export 3D Model, 120  
Export CIF File, 319  
Exporting  
    a 3D model, 120, 180, 378  
    a CIF file, 319  
    a netlist, 38  
Extracting  
    a netlist, 70

## F

FILL deposit, 199, 632  
Folded spring  
    generation, 61  
    instantiation, 23  
Folded spring, 564

Foundry Support, 8

## G

Generated layout parameters  
    edition, 494

Generating  
    a 3D model, 77  
    a comb-drive, 59  
    a ground plate, 62  
    a plate, 58  
    bonding pads, 63  
    folded springs, 61

Global nodes  
    connection, 27

Ground plate  
    generation, 62

Ground plate, 566

Grow, 657

Guckel ring  
    array, 554

Guckel ring test structure, 551

## H

Harmonic side drive, 517

Help menu, 131

Holes

    creation, 121

## I

ImplantDiffuse, 652

Import MEMS, 302, 345

Importing

    a 3D model in ANSYS, 380

    MEMS, 302, 345

    process definition, 186

Instantiating

    a plate, 21

    comb-drives, 22

    components, 21

    folded springs 23

    voltage sources, 27

Interpolation, 417



## J

Journal bearing  
    type 1, 520  
    type 2, 523

## L

Layout  
    extraction, 97  
LAYOUT button, 320  
Layout Versus Schematic  
    Voir LVS  
L-Edit, 5  
L-Edit/UPI, 55  
Library menu  
    Edit component, 112  
    library palette, 110  
Library menu, 110  
Library palette  
    editing parameters, 494  
Library palette, 110

Library, 8  
Linear crab-leg  
    type 1, 526  
    type 2, 529  
Linear Electrostatic comb-drive, 496  
Linear folded beam, 532  
Linear side drive, 499  
LVS  
    extracting schematic, 102  
    launching 105  
LVS, 6, 105

## M

Macros  
    alignment functions, 577  
    all angle objects approximation, 594  
    concentric circles generation, 598  
    logo generator, 574  
    plate release, 584  
    polar array, 580  
    viewing vertex coordinates and angles, 587

Material properties  
    setting in ANSYS, 382

MCNC MUMPs  
    cross-section, 605

MCNC MUMPs, 604

MechanicalPolish, 209, 647

MEMS  
    import, 302, 345

MEMS Layout Palette  
    using, 57

MEMS layout palette 56  
    Active Elements tab, 55  
    Passive Elements tab, 55  
    Resonator Elements tab, 55  
    Test Elements tab, 55

MEMS Library, 8

MEMS Pro  
    tool flow, 3

Meshing  
    3D model in ANSYS, 388

Module  
    creation, 20

MOSIS/CMU, 609

MOSIS/NIST, 610  
Multilayer pad, 557

## N

### Netlist

- comparison, 105
- export, 38
- extraction, 70
- simulating from, 40

### Nodes

- labeling, 32

## O

### Object properties

- edition, 30

### Optimization

- examining the output, 467
- running, 466
- setting up, 453

Optimization, 450

Optimizing

    a design, 450

## P

Pad

    bonding, 568

    multilayer, 557

Plate

    generation, 58

    ground, 566

    instantiation, 21

Plate Release, 121

Plate, 559

Polar Array, 123

Polygon

    drawing (curved), 90

Ports

    about, 66

Process

    edition, 663

Process definition  
    edition, 116, 185  
    import, 73, 186  
    process steps, 616  
Process definition, 611  
Process step parameter  
    Deposit, 194, 622  
    Etch, 201, 635  
    Grow, 657  
    ImplantDiffuse, 652  
    MechanicalPolish, 209, 647  
    ProcessInfo, 616  
    Wafer, 191, 618  
Process steps  
    edition of parameters, 190  
    edition, 186, 188  
Properties  
    view, 69  
Property  
    creation, 407

## R

R.O.M menu

3D To Layout menu, 298

R.O.M menu, 218

Reduced Order Modeling

Voir ROM

Reduction algorithm, 235, 281

Results

viewing in ANSYS, 392

ROM

condensation algorithm, 222

reduction algorithm, 235

ROM, 216

Rotary comb-drive, 511

Rotary motor, 136

Rotary side drive, 514

Routing

a design, 443

## S

SACRIFICIAL etch type, 202, 208

SAT file format, 144

Save MEMS, 317

Saving

MEMS, 317

Schematic

creation, 17

extraction, 102

Schematic symbol

creation, 472

Schematic tools

Command tool, 400

S-Edit, 4

Show Details button, 492

Side drive

harmonic, 517

linear, 499

rotary, 514

Simulation

from a netlist, 40

using T-Spice, 40



Simulation, 33  
SNOWFALL deposit, 197, 630  
Spice models, 475  
Splines  
    approximation, 420  
    creation, 125, 415  
    drawing, 95  
    edition, 127, 429  
    interpolation, 417  
    understanding, 410  
Splines menu 125  
    Create, 125  
    Edit, 127  
Spring  
    dual archimedean spiral, 535  
    folded, 564  
Spring, 377  
SURFACE etch type, 202

## T

Technology

Analog Devices/MCNC iMEMS, 607  
MCNC MUMPs, 604  
MOSIS/CMU, 609  
MOSIS/NIST, 610  
Sandia ITT, 608  
Technology file, 328  
Technology setup, 602  
Test structure  
    area-perimeter dielectric isolation, 538  
    crossover (1), 541  
    crossover (2), 543  
    Guckel ring, 551  
Thermal actuator, 134  
Tool Flow 3  
Tools menu  
    Clearing Vertex Information, 130  
    Viewing Vertex Angles, 129  
    Viewing Vertex Coordinates, 128  
    Viewing Vertex Information, 129  
Tools menu, 128  
Torus  
    drawing, 88  
T-Spice

launching 39  
    simulation, 40  
    user interface, 40  
T-Spice Pro, 4

## U

Unidirectional rotary comb-drive  
    type 1, 502  
    type 2, 505  
UPL, 6  
Utilities, 570

## V

Vertex  
    clearing information, 130  
    viewing angles, 129  
    viewing coordinates, 128  
    viewing information, 129  
View 3D Model, 118

## Viewing

- a 3D model in ANSYS, 380
- a 3D model, 72, 81, 118, 149
- a cross-section, 175
- analysis results, 392
- properties, 69
- vertex angles, 129
- vertex coordinates, 128
- vertex information, 129
- waveform, 45

## Viewing Vertex Angles, 129

## Viewing Vertex Coordinates, 128

## Viewing Vertex Information, 129

## Voltage sources

- instantiation, 27

## Volumes

- addition, 310, 356
- creation, 305, 353
- deletion, 309, 360

## W

Wafer, 191, 618

Waveform

    probing, 43

    viewing, 45

Wire

    drawing, 87

Wire tool, 24